AD-A039 500 WASHINGTON UNIV SEATTLE DEPT OF OCEANOGRAPHY F/G 9/2
WHIMPER--A LIBRARY OF TIME SERIES PROGRAMS UNIVERSITY OF WASHIN--ETC(U)
DEC 76 J D IRISH, D J BENDINER, M D LEVINE N00014-67-A-0103-0014

UNCLASSIFIED TR-342 NL

| OF |

AD
A039500

END

DATE
FILMED
6-77

the department of

OCEANOGRAPHY

(12)

Technical Report No. 342

WHIMPER

A Library of Time Series Programs
University of Washington
For use on CDC 6400 Computer

Office of Naval Research
Contract N00014-67-A-0103-0014
Project NR 083 012

Reference M76-105
December 1976

seattle, washington 98195

UNIVERSITY OF WASHINGTON
DEPARTMENT OF OCEANOGRAPHY
Seattle, Washington 98195

Technical Report No. 342

WHIMPER

A Library of Time Series Programs
University of Washington
For use on CDC 6400 Computer

James D. Irish
Donna J. Bendiner
Murray D. Levine
Judy Zeh

Office of Naval Research
Contract N-00014-67-A-0103-0014
Project NR 083 012

Reference M76-105, A.P.L. 7508
December 1976

Francis A. Richards
Principal Investigator and
Associate Chairman for Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: (9) Technical Report No. 342 — 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): (6) WHIMPER--A Library of Time Series Programs University of Washington, For use on CDC 6400 Computer	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER: M76-105, A.P.L. 7508
7. AUTHOR(s): (10) J. D. Irish, D. J. Bendiner, M. D. Levine, ~~and~~ J. Zeh (Donna, Murray, Judy)	8. CONTRACT OR GRANT NUMBER(s): N-00014-67-A-0103-0014
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Washington, Department of Oceanography & Applied Physics Lab, Seattle, Washington 98195	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project NR 083 012
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, La Jolla, California	12. REPORT DATE: (11) December 1976
	13. NUMBER OF PAGES: 73 (12) 79p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

(14) TR-342, Ref-M76-105

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

370 280

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Oceanographic Research
Computer programs
Time series programs
CDC 6400 computer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Documentation is ~~given~~ provided for the use of computer subroutines for data analysis on the University of Washington campus computer. This includes routines for data storage and retrieval, spectral analysis, filtering, and tidal analysis.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This is the way the world ends
this is the way the world ends
this is the way the world ends,
not with a bang but a whimper.

The Hollow Man
T. S. Eliot

The authors wish to thank Walter Munk and Florence Dormer for the tide routines and some other routines which have been adapted from their BOOM library of time series programs. Also persons in the Department of Oceanography, University of Washington for comments on these routines.

Use of Whimper Library

The library is maintained by Jim Irish on permanent file on the campus CDC computer. Program listings are available at the Applied Physics Laboratory, University of Washington. To use, the following control cards should be used:

```
ATTACH (WHIMPER, ID = IRISH)
  .
  .
  .
LDSET (LIB = WHIMPER)
LGO.
```

For certain programs, the Computer Center GRAFIX library must also be attached. Programs requiring this additional library are LGAX, CCCOH and PLOTDR. If a Cal. Comp plot is desired, a dispose card is required.

```
ATTACH (WHIMPER, ID = IRISH)
ATTACH (GRAFIX)
DISPOSE (TAPE99,*CC)
  .
  .
  .
LDSET (LIB=WHIMPER/GRAFIX)
LGO.
```

INDEX

GEOTAPE Data Archiving System

The GEOTAPE system was designed for archiving geophysical data. The concept was borrowed from the system used by Walter Munk at the Institute of Geophysics and Planetary Physics at San Diego. The GEOTAPE system is a method for archiving, on magnetic tape or disk, any kind of geophysical data that can be reduced to integer form. A main feature is the incorporation of documentation as part of the record for each data file. Each file becomes a self-sufficient document containing such information as: how the raw data was taken and how it has been converted, edited, digitized or modified to result in the present values.

The overall structure of a GEOTAPE is a series of files recorded in standard BCD code. The first file on a tape is an ID file and the remaining files are the data files written in GEOTAPE format. Each GEOTAPE data file consists of a "legend" record followed by a series of data records. The legend record is in the form of 80-character card images and contains pertinent information about the data in the remaining records of the file. The user may include as much (limited to 64 card images) or as little information as desired in the legend record; the only restriction is the last card image must be a special ENDLEGEND card. This card has a specific format and provides required information about the data matrix, e.g. the format of the data records, the number of data values, etc. For more details, see ENDLEGEND card format below.

The data records of the file are in the form of 120-character line images. Characters 1-100 contain the integer data values, characters 101-110 a checksum, and characters 111-120 an ID word specifying file number, record number, and line number. There are 30 of these line images per data record (a total of 3600 characters/record). The last data record of the file can be shorter than 30 lines. If the last data value of the set falls in the middle of a line image, the remaining positions in the 100 data characters are filled with 9's to denote the the end of data.

The ID file for a GEOTAPE contains only one record, and its format is similar to the legend record of a data file. The record is a series of 80-character card images which gives information about all data files on the tape, such as: when and where the data was taken, who to contact for information, etc. As with the legend record of a data file, the user may provide as much (limited to 64 cards) or as little information as desired in the tape ID file. The only requirement is that the last card image must be a special ENDIDFILE card. For details see ENDIDFILE card format below. When data is being added to a GEOTAPE file by file, the user may want to collect only data files on the tape, and then add the tape ID file at a later time, when the data files are finalized. If, however, a user is archiving data taken in the past and knows exactly how many files he will store on a GEOTAPE, it may be convenient to write the tape ID file first and then the data files. Lack of a tape ID file on a GEOTAPE does not cause a fatal error or a halt in any of the GEOTAPE programs, although it does cause a warning to be printed in the program GEOPRNT. A new ID file may be inserted at the beginning of a GEOTAPE with COPYGEO.

A series of FORTRAN programs has been written for storing and retrieving data in GEOTAPE format. The following general purpose routines comprise the software for the system:

ARCHIVE = subroutine which archives a set of integer data and writes a file in GEOTAPE format;

RETREVE = subroutine which reads and checks data from a GEOTAPE file and makes it available in an integer array or on a scratch disk file;

COPYGEO = program which copies a series of GEOTAPE files from one disk or tape to another; can also create a new ID file for a GEOTAPE:

GEOPRNT = program which lists records from a series of GEOTAPE files with a choice of three printing options.

Brief List of Restrictions:

1. Programs in system are written in FORTRAN IV for use on a CDC 6400.
2. All GEOTAPE files are written in standard BCD code and 7-track tape is assumed when magnetic tape is the output medium.
3. All data to be recorded on a GEOTAPE file must be in integer form.
4. The "legend" record is limited to 64 card images.
5. The last card of the "legend" record has a fixed format. See ENDLEGEND card format for details.
6. The tape ID file record is also limited to 64 card images, with the last card having a fixed format. See ENDIDFILE card format for details.
7. The data records of a file are limited to 30 line images with 100 characters devoted to data values in each line.
8. The data matrix recorded for each file can have at most two dimensions. Three-dimensional matrices are not handled, or must be treated as one or two-dimensional.
9. Because of the format for the ID word of each line image, the number of files per GEOTAPE is limited to 269 and the number of data records per file cannot exceed 999.

ENDLEGEND Card Format and the Legend Record

The legend record is the first record of each GEOTAPE data file. It contains the documentation for the data in the file and provides sufficient information for a user to read the data records and interpret them properly. The legend record is in the form of 80-character BCD card images, with the first character of each card always blank (for carriage control). The size of the record is limited to 64 card images (or 5120 characters), and the last card image must be the ENDLEGEND card in the following format:

Example:

```
2            15    20   24     30 33 36 39 42       50         55-70            71-80
|             |     |    |      |  |  |  |  |        |
ENDLEGEND   112   540   9X   6700(18I05,10X,I10,A10/)    644876.9333333    0.0333333
```

Cols.	Explanation
1	must be blank
2-10	the word 'ENDLEGEND'
11-15	number of data records in this file (right-adjusted)--obtained by dividing total number of data points (cols. 21-23 times cols. 25-30) by number of data points per record (cols. 16-20) and adding 1 for any remainder.
16-20	maximum number of data points per record (right-adjusted)--obtained by multiplying number of data points per line (cols. 32-33) by 30 lines per record.
21-23	the first dimension M (right-adjusted) if the data array is two-dimensional--M usually indicates number of sensors--should be '1' if array is one-dimensional.
24	the letter 'X'
25-30	the second dimension N (right-adjusted) of the data array, forming an M x N matrix--N usually indicates a point in the time domain--if array is one-dimensional, M will be '1' and N will be the total number of data points for this file.

Cols.	Explanation
31-50	format for reading each line image of data in the file, including:

Cols.	
32-33	number of data points per line (right-adjusted).
35-36	number of digits per data point, counting the sign as one digit (right-adjusted).
38-39	number of spaces to be skipped from last data value of line to get to col. 100 of line image--leave cols. 38-41 blank if no spaces are to be skipped.
42-45	format for checksum (I10).
46-48	format for ID word (A10). The word is of the form FxxRyyLzz, where xx is the file number, yy is the the record number, and zz is the line number--for the file number only, when xx>99, the first digit can be A-Z (then, xx=Z9 representing file #269 is the largest file number).

55-70	start time of data measurements in hours from the turn of the century.*
71-80	sample interval in hours.*

This format for the ENDLEGEND card must be followed exactly if the user intends to make use of the subroutines ARCHIVE and RETREVE.

---

* Actually, these fields may be used to store any two numbers which are important to the user. These two numbers will be returned to the calling program by subroutine RETREVE via the variables TSTART and TDELT. If no decimal points appear in these fields, the format is assumed to be F16.7 for cols. 55-70 and F10.7 for cols. 71-80.

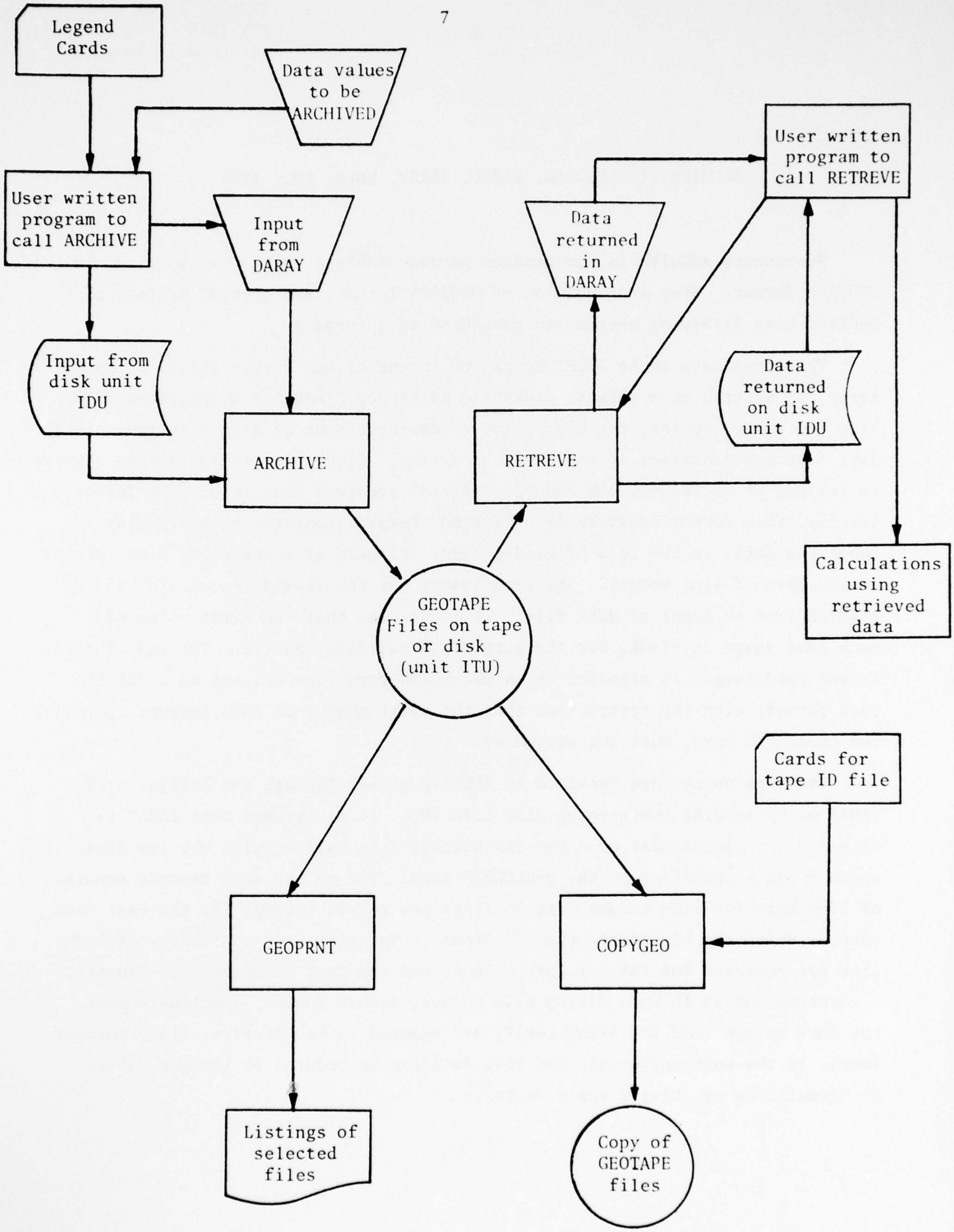
ENDIDFILE Card Format and Tape ID File

The tape ID file contains only one record, which is similar to the legend record of a data file, and contains any necessary information about the tape, such as: how many data files it contains, who to contact for more information, etc. This record is in the form of 80-character BCD card images, with the first character of each card being a blank. The size of the record is limited to 64 card images, and the last card image must be the ENDIDFILE card in the following format:

Example:

```
2                  21          30   35
|                   |           |    |
ENDIDFILE           GEOTAPE 13     76 DATA FILES
```

Cols.	
1	Must be blank
2-10	the word 'ENDIDFILE'
21-27	the word 'GEOTAPE'
28-30	the identifying number for this GEOTAPE (right-adjusted)
31-35	number of data files on this tape (right-adjusted)
36-80	any explanatory comments


Legend Cards
Data values to be ARCHIVED
User written program to call ARCHIVE
Input from DARAY
Input from disk unit IDU
ARCHIVE
User written program to call RETREVE
Data returned in DARAY
Data returned on disk unit IDU
RETREVE
Calculations using retrieved data
GEOTAPE Files on tape or disk (unit ITU)
Cards for tape ID file
GEOPRNT
COPYGEO
Listings of selected files
Copy of GEOTAPE files

ARCHIVE (IFILE, ILU, IAØRD, IARAY, NBLK, IDU, ITU)

Subroutine ARCHIVE is the general purpose routine which creates files in GEOTAPE format. (For a discussion of GEOTAPE format, see general writeup of GEOTAPE Data Archiving System and ENDLEGEND card format).

The input data to be ARCHIVEd can be in one of two forms--stored in an array, or written on a scratch disk file in binary blocks of a specified size. With the latter option, the subroutine reads the blocks of data from the scratch disk file and processes it one block at a time. Each time the subroutine ARCHIVE is called, it writes out one complete file of archived data in GEOTAPE format. Briefly, this format consists of a "legend" record (containing information about the data) in the form of card images, followed by a series of data records in the form of line images. The card images for the legend record are read by ARCHIVE from an input or disk file. It is assumed that the first colum of each card image is blank, for the purpose of carriage control. The end of these legend card images is signaled by an ENDLEGEND card (see writeup on ENDLEGEND card format) with the restriction that the total number of card images, including the ENDLEGEND card, must not exceed 64.

The data values are received by ARCHIVE either through the integer array IARAY or by reading the scratch disk file IDU. It is assumed that IARAY is either a one-dimensional or a two-dimensional data matrix, with the two dimensions M and N specified on the ENDLEGEND card. The output data records consist of 120-character line images with 30 lines per record (except for the last data record, which can be shorter than 30 lines). The first 100 characters of each line are reserved for integer data values, and the last 20 characters contain a checksum and an ID word giving file number, record number, and line number. The data values (and the array IARAY) are assumed to be integers, since integer format is the most universal, and most data can be reduced to integer values by normalizing or using a scale factor.

Input:

IFILE = Geotape file number for this set of data (will be written as part of ID word for each line image);

ILU = unit number from which legend card images are to be read for this file; it is assumed that unit ILU is positioned correctly for ARCHIVE to read the legend card images from it;

IAORD = a flag which indicates whether data is transmitted through IARAY (IAORD = +1) or has been recorded on a scratch disk file IDU (IAORD = -1);

IARAY = integer array used for storing the data values; total dimension in calling routine must be at least NBLK; the matching array to IARAY in the calling parameters can be two-dimensional, but the first dimension M must correspond to the M (first) dimension on the ENDLEGEND card;

NBLK = size of blocks written on scratch disk file IDU when IAORD = -1; when IAORD = +1 NBLK must equal M*N from the ENDLEGEND card (if it does not, M and N on the ENDLEGEND card are reset to 1 and NBLK, respectively);

IDU = unit number of scratch disk file where data is recorded when IAORD = -1; when IAORD = +1, it doesn't matter what IDU is; unit IDU is automatically rewound at the beginning of ARCHIVE only when IAORD = -1;

ITU = unit number for magnetic tape or disk output; it is assumed that this unit is already correctly positioned for the writing of this particular data file; no rewinding or repositioning of unit ITU is done in subroutine ARCHIVE, except for the actual writing of the data file.

In the case of the input scratch disk file (IAORD = -1) it may be that the total number of data points (M*N) is not an exact multiple of NBLK. ARCHIVE will automatically do this calculation, decide whether the last disk block has fewer than NBLK words, and read it correctly.

Similarly, it may be that the total number of data points (M*N) is not an exact multiple of NVALREC, the number of data values per record. In this case, the final record which is written will be shorter than 30 line images. If the last data value falls at the middle of the last line image, the remainder of the line is filled with 9's, although the 9's are not added in when the checksum for the line image is calculated.

It should be remembered that all three disk or tape units--ILU, IDU, and ITU--should be declared in the PROGRAM card of the mainline. Unit IDU need not be declared, however, if it is not to be used.

RETREVE (IFILE, IAORD, IARAY, M1, M2, N1, N2, NBLK, IDU, ITU, M, N, TSTART, TDELT, IERR)

Subroutine RETREVE reads a block of data from a GEOTAPE and performs certain checks on the data file (for a discussion of GEOTAPE format, see general writeup of GEOTAPE Data Archiving System and description of ENDLEGEND card format).

The data to be RETREVEd from the GEOTAPE file can be returned to the calling program in one of two ways--stored in the array IARAY, or written on a scratch disk file IDU in blocks of size NBLK. You can request the data you want from the file by setting the formal parameters M1, M2 and N1, N2 in the subroutine call. Matrix columns M1 through M2 and rows N1 through N2 will then be RETREVEd from the total data matrix. Of course, these four numbers--M1, M2, N1, N2--cannot exceed the total dimensions of the data matrix, as indicated by M and N on the ENDLEGEND card, and the subroutine checks them after reading the legend to make sure. The default case for data retrieval is the whole data matrix. This can be requested simply by setting M1 = 0 and N1 = 0 in the call to RETREVE. Also, all the columns of the matrix can be requested for a particular set of rows by setting M1 = 0, and similarly all rows are retrieved with N1 = 0. So, if you wanted to RETREVE all the rows of a data matrix for columns 2 through 6, although you had no idea how many rows there were, you could set M1 = 2, M2 = 6, N1 = 0 and specify the output to be recorded on a scratch disk file by setting IAORD = -1, IDU = some declared unit number, and NBLK = some convenient blocking size, say 1000. RETREVE would then return in M and N the actual number of columns and rows retrieved, and the calling program would be able to compute how many blocks of 1000 words each to read from unit IDU.

Each time RETREVE is called, it assumes input unit ITU is positioned correctly at the beginning of file IFILE, and then, after retrieving the requested data, it skips over remaining data records in the file and over the EOF, so that unit ITU is always positioned at the beginning of the next file when control is returned to the calling program.

Input:

IFILE = file number of the file which data is to be retrieved; this is checked against the file designation in the ID word, and an error message is printed if they do not agree;

IAORD = flag which indicates whether the data retrieved from this data file is to be stored in the array IARAY (IAORD = +1) or to be written on a scratch disk file in blocks of size NBLK (IAORD = -1);

IARAY = integer array used to store retrieved data (dimension is at least NBLK). When IAORD = +1, the matching array in the calling sequence can be one-dimensional or two-dimensional, but the dimensions of this array must be large enough to hold all the requested data (M2-M1+1 by N2-N1+1);

M, N = the number of columns and rows retrieved and stored in IARAY or on the scratch disk file; normally, M=M2-M1+1 and N=N2-N1+1 will be returned, but in the default case, M=M (on ENDLEGEND card) and N=N (on ENDLEGEND card) will be returned;

M1, M2 = indicates which columns of the data matrix are requested; columns M1 through M2, inclusive, are retrieved out of the total 1 through M (from the ENDLEGEND card); when M1=0 is encountered, the default option is assumed, namely, columns 1 through M are retrieved;

N1, N2 = indicates which rows of the data matrix are requested; rows N1 through N2, inclusive, are retrieved out of the total 1 through N (on ENDLEGEND card); when N1=0 is encountered, the default option is assumed, namely, rows 1 through N are retrieved;

NBLK = size of data blocks to be written on scratch disk file IDU when IAORD = -1; NBLK is also the dimension of IARAY, so when IAORD = +1, NBLK should be at least the total number of points to be retrieved in IARAY;

IDU = unit number of scratch disk file on which retrieved data is written when IAORD = -1; it is the responsibility of the calling program to rewind this disk file when it is used both before and after each call to RETREVE;

ITU = unit number of magnetic tape or disk from which GEOTAPE file data is read.

TSTART = sample interval of data measurements in hours (also from ENDLEGEND card)*;

TDELT = sample interval of data measurements in hours (also from ENDLEGEND card)*;

IERR = error flag which is set=0 when no fatal error has occurred in RETREVE, and is set=+1 when a fatal error has occurred; when RETREVE returns an IERR=+1, it still positions unit ITU at the beginning of the next file, but the status of IARAY or disk file IDU is uncertain.

No matter whether RETREVE returns the full data matrix, only part of the data, or finds an error in the first record, it still reads and checks every record in the file and positions unit ITU at the beginning of the next file. Most importantly, it sums the data values for every line image in the file to see whether the sum agrees with the checksum recorded for this line. This feature provides a constant check on much-used data tapes and tells the owner when the tape is getting worn or old and should be copied to a new reel. The checking of the ID word (which is done only for the first data record of the file) provides an automatic comparison between where the tape is actually positioned and the position from which the user wants the data retrieved. Only if the ID file designation agrees with IFILE, and only if the record and line designations indicate the appropriate line of the first record, will RETREVE continue to read the data records. The first record of the file should be the legend record; an error message is printed if it is not, and +1 is returned in IERR. The printed output of RETREVE consists of the legend record plus a short message telling how many rows and columns were retrieved from the data matrix.

---

* Actually, any two important numbers can be stored in these positions on the ENDLEGEND card and will be returned by RETREVE in TSTART and TDELT.

It should be remembered that unit ITU should be declared in the PROGRAM card of the mainline, and unit IDU should also be declared if it is to be used in RETREVE.

COPYGEO (INPUT, OUTPUT, TAPE1, TAPE2)

COPYGEO is the general purpose program which copies a series of files in GEOTAPE format from one disk or tape unit to another (for a discussion of GEOTAPE format, see general writeup of GEOTAPE Data Archiving System and ENDLEGEND card format).

The program is designed to copy both kinds of GEOTAPE files--tape ID files and data files. It also has an option to create a new tape ID file from cards. Very little checking is done when each file is copied. A fatal error occurs only when the first record of each file is not either a legend record or the ID file record. For a data file, NREC (number of remaining data records in the file) is read from the legend record and checked against the actual count of records read for the file, and a warning is issued if it does not agree. In the case of a new ID file being created, the card images are checked to make sure that the total number of cards does not exceed 64, and that the last card is an ENDIDFILE card.

The unit numbers for input and output are fixed in COPYGEO. Card images for a new ID file are read when IFIDNEW = +1. TAPE2 is the input unit for the GEOTAPE files to be copied, and TAPE1 is the output unit. TAPE2 and TAPE1 are assumed to be disk or tape units.

Three input parameters must be supplied to COPYGEO by the user format(3I5):

IFIDNEW - a flag which indicated whether there is a new ID file to be copied to TAPE1 (IFIDNEW = +1); when there is no new ID file, IFIDNEW = -1;

NFSKIP - number of files to be skipped on TAPE2 before the copying begins; NFSKIP = 0 causes no files to be skipped before copying;

NFCOPY - number of files in GEOTAPE format to be copied from TAPE2 to TAPE1; NFCOPY = 0 implies the default option, which is to copy all files on TAPE2 to TAPE1 until an EOI is encountered.

The above three parameters are read in on the first data card in the format (3I5). An additional fourth parameter is read in from a second card only when IFIDNEW = +1:

TAPNAME - 10-character name for the tape; used for the purposes of printout only; read on a second data card in the format (A10).

New IDLEGEND cards follow when IFIDNEW = +1.

It is assumed by COPYGEO that the input unit--TAPE2--is properly rewound and/or positioned before execution. TAPE2 should be positioned so that COPYGEO can skip and then copy the files specified by NFSKIP and NFCOPY on the first data card. For a discussion of the format of a tape ID file, see general writeup of GEOTAPE Data Archiving System. It is assumed that the first character of each card image in the ID file is a blank.

GEOPRNT (INPUT, OUTPUT, TAPE1)

GEOPRNT is the general purpose program which lists files in GEOTAPE format with a choice of three printout options, specified by the input variable IOPT. IOPT = 1 produces the least amount of printout per file and IOPT = 3 produces the most.

GEOPRNT can handle both ID files and data files, but it does assume that the first record of each file is either an ID file record or a legend record and prints a warning message if it is not. TAPE1 is the input disk or tape unit which contains files in GEOTAPE format for printing. Three input parameters must be supplied to GEOPRNT by the user: FORMAT (3I5)

IOPT - one of three printout options;
IOPT = 1 prints only the legend record of each file;
IOPT = 2 (the default) prints the legend record plus the first and last data records of each file;
IOPT = 3 prints the legend record plus all data records in each file.

NFSKIP - number of files to be skipped on TAPE1 before listing of files begins; NFSKIP = 0 causes no files to be skipped before listing;

NSPRNT - number of GEOTAPE-formatted files to be listed according to option IOPT; NFPRNT = 0 implies the default, which is to list all files on TAPE1 until an EOI is encountered.

If a value is read for IOPT which is anything other than 1, 2, or 3, it is reset to the default (IOPT = 2). If there is a file to be listed on TAPE1 which does not have a legend record, the first record (which is always printed for every file) will not be formatted correctly. In this case, if IOPT = 2, GEOPRNT will not be able to read NREC from an ENDLEGEND card and will not be able to print the last data record of the file. GEOPRNT does little or no checking of the data files on TAPE1; it attempts merely to list whatever is there.

Example of ARCHIVE

```
GEOTAPE, P1, IØ100, T240, MT1.
ACCOUNT (----)
FORTRAN (LC = 15000)
   .
   .
ATTACH (TAPE1, DATA, ID = INTERGERIZED)
REQUEST (TAPE2, VSN = P41, HI, OUTPUT)
SKIPF (TAPE2, 1, 17)
DISPOSE (OUTPUT, *PR = C)
ATTACH (WHIMPER, ID = IRISH)
LDSET (LIB   WHIMPER)
LGO.
EOR
       PROGRAM GEOTAPE (INPUT,OUTPUT,TAPE5=INPUT, TAPE1,TAPE2)
       DIMENSION IARAY (4,2832)
       PRINT 10
   10  FORMAT (1HT)
       CALL ARCHIVE (2,5,-1, IARAY, 11328, 1,2)
       STOP
       END
EOR
   legend cards
EOF
```

The disk file (data) containing the integerized data is attached, the output tape (P41) requested and positioned at file 2 where the data is to be stored. The legend is read from data cards. The printed output from ARCHIVE contains all the data written on tape. This output is printed at the central site at 8 lines per inch. The IØ and T parameters on the job card, and the LC parameters on Fortran card must be specified since they are generally greater than the default values. The may be estimated by,

Lines of output, LC = 5 + number of legend cards + number of data values divided by the number per line + number of records in this file. (The last two numbers can be obtained from the ENDLEGEND card.)

CP Time required, T = 0.02 seconds times the lines of data.

IO Time required, IØ = 0.005 seconds times the lines of data.

Example of RETREVE

The program retrieves a geotape record from magnetic tape, normalizes the integer counts to degrees centigrade and stores the results in permanent file.

```
RETRIEVE, MT1, T30.
ACCOUNT (----)
FORTRAN.

REQUEST (TAPE1, VSN = P784, HI, S, INPUT)
SKIPF (TAPE1, 1, 17)
REQUEST (TAPE2, *PF)
ATTACH (WHIMPER, ID = IRISH)
LDSET (LIB = WHIMPER)
LGO.
CATALOG (TAPE2, TEMP, ID = BOTTOM)
EOR
      PROGRAM GET (INPUT, OUTPUT, TAPE1, TAPE2)
      DIMENSION IARAY (2048), TEMP(2048)
      EQUIVALENCE IARAY(1), TEMP(1)
      CALL RETREVE (2,1, IARAY, 2, 2, 3, 2050, 2049, 0, 1, M, LOS,ST,DT, IE)
      IF(IE·NE·O) GO TO 999
      DO 100 I = 1, LOS
  100 TEMP(I) = 1.E-04*FLOAT(IARAY(I))
      WRITE (2) TEMP
  999 STOP
      END
EOF
```

The requested tape 1 is a GEOTAPE written at 556 BPI and must be declared a stranger tape, S, at HI density. RETREVE could have stored the data in disk file by setting the second parameter negative, instead the numbers are scaled to be floating point in degree centigrade. The program retrieves the second of the M possible merged series and only asks for terms 3 through 2050 inclusive.

Example of COPYGEO and GEOPRNT

This example copies tape 2 to tape 1, adding a new IDFILE, then listing the new tape with GEOPRNT.

```
TAPCOPY, MT2.
ACCOUNT (----)
REQUEST (TAPE2, VSN = P784, INPUT)
REQUEST (TAPE1, VSN = P41, HI, OUTPUT)
SKIPF (TAPE2, 1, 17)
ATTACH (WHIMPER, ID = IRISH)
LDSET (LIB = WHIMPER)
COPYGEO.
REWIND (TAPE1)
GEOPRNT.
EOR
....1....0....0
GEOTAPE 12
(new IDFILE cards)
EOR
....1....0....2
EOF
```

The input tape 2 is a scope tape, and the output is a HI density copy. The first file, the old IDFILE, is skipped and a new IDFILE read from cards. When the copy is made the tape 1 is rewound, and the data listed by the use of GEOPRNT.

GNCOS (SERIES, LOS, FREQ, AMP, PH, N,IP)

Function GNCOS generates a sum of cosines, SERIES, at frequencies specified in FREQ, with amplitudes in AMP and phase in PH.

$$SERIES_k = \sum_{j=1}^{N} AMP_j \cdot COS(\pi \cdot FREQ_j (k+IP)+PH_j), \; k=0, 1,..., LOS-1$$

Input:

LOS = number of points in resultant cosine series (the dimension of SERIES.

FREQ = array of dimension N containing the component frequencies in Nyquists, where 1 Nyquist = 1/(2•DT) where DT is the time interval between terms of SERIES. Note the frequency in harmonics is just (2/LOS) times the frequency in Nyquists.

AMP = array of dimension N containing the component amplitudes.

PH = array of dimension N containing the component phases.

N = number of components in the sum of cosines. Also the dimension of FREQ, AMP, and PH.

IP = initial phase of Sum. This is normally set zero. If the series is to be generated in pieces then IP = 0 for the first call, then equals the number of terms in the first piece for the second call, etc. It adds a phase = $\pi \cdot FREQ \cdot IP$ to the arguement of the cosine of each term.

Output:

SERIES = array of dimension LOS containing the desired sum cosines.

The function sets GNCOS = 0. if the call was successful

= 1. if LOS or N $\leq$ 0.

GNFIL(FILTER, M, IC, IA, IB, F)

Function GNFIL generates a low or high pass filter for use as a lag window, as a data window, or as the impulse response of a linear system. Consider the following:

In real space

Linear System

$$x(t) \xrightarrow{\text{input}} \boxed{\begin{array}{c} h(t) \\ \text{impulse response} \end{array}} \xrightarrow{\text{output}} y(t) = x(t)*h(t) \quad \text{(convolution)}$$

In fourier space

$$X(\omega) \xrightarrow{\text{input}} \boxed{\begin{array}{c} H(\omega) \\ \text{frequency response} \end{array}} \xrightarrow{\text{output}} Y(\omega) = X(\omega)\cdot H(\omega) \quad \text{(multiplication)}$$

where $x(t):X(\omega)$, $y(t):Y(\omega)$, and $h(t):H(\omega)$ are Fourier Transform pairs.

The series FILTER is the digital analog of $h(t)$ and is designed by examining the desired properties of $H(\omega)$. (Fig. 1 and 2) See example.

Input:

M = length of right half of filter. Dimension of FILTER depends on IA.

IC determines the type of filter

= 1 - TRIANGULAR taper

= 2 - PARZEN taper

= 3 - COSINE taper

= 4 - LANZCOS taper

= 5 - LANZCOS SQUARED taper

IA = 0, produces a one-sided series of M+1 terms fading to the right for use with NCONVL. The total filter is a symmetrical reflection about the center term.

= 1, prodces a symmetrical series of 2*M+1 terms.

IB determines the normalization

= 0 produces filter with no normalization. The magnitude of the largest term is 1.

= 1 produces a low-pass filter normalized so the sum of terms equals the number of terms.

= 2 produces a high-pass filter with response complementary to that obtained with IB = 1. The normalization is such that the sum of terms is zero.

F determines the cutoff frequency. If F $\neq$ 0, GNFIL returns a filter whose transmission is 6 db down (1/2 amplitude) at a frequency of F Nyquists. (1 Nyquist $\equiv$ 1/(2*DT), where DT is the time interval between terms.)

Output:

FILTER - Array containing filter weights.

The function sets GNFIL = 1 if IA < 0 or IA > 1
2 if IB < 0 or IB > 2
3 if IC < 0 or IC > 5
0 otherwise

To create a filter:

1. Choose either low-pass (IB=1) or high-pass (IB=2) filter (Figure 1.)
2. Choose the cutoff frequency F in Nyquists
3. Choose the rate of energy fall-off by specifying the energy rejection at a frequency F+q Nyquists. Use Figure 2 to find the corresponding value of Mq and determine M. Generally a larger value of M means a sharper rolloff in energy rejection of the filter.

Example:

1. Choose a triangular, low-pass filter, so IB=1 and IC=1.
2. Take F for 6 db down to be 1 cph. The sample interval is 1 minute so the Nyquist frequency is 30 cph. So F=0.03333.
3. Want 40 db energy rejection at 5 cph so q=0.1333333. From Figure 2 Mq=10 for 40 db rejection with IC=1, so M=75.

Therefore, the call would be

G = GNFIL(FILTER, 75, 1, 0, 1, 0.0333333)

If a LANZCOS filter, IC=4, were used. Then for 40 db energy rejection Mq = 1.8 so M = 14, and the filter length for a given energy reduction is reduced by a factor of 5. The convolution with this filter will be much faster. The call would be

G = GNFIL(FILTER, 14, 4, 0, 1, 0.0333333)

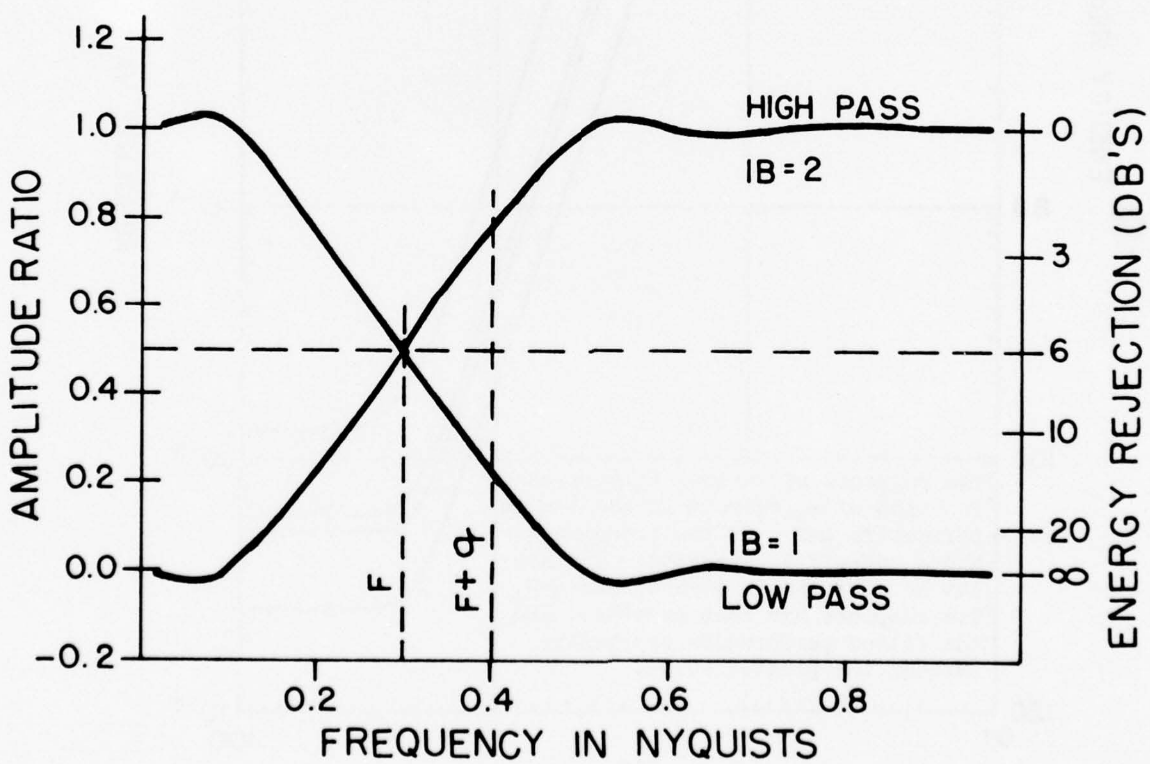
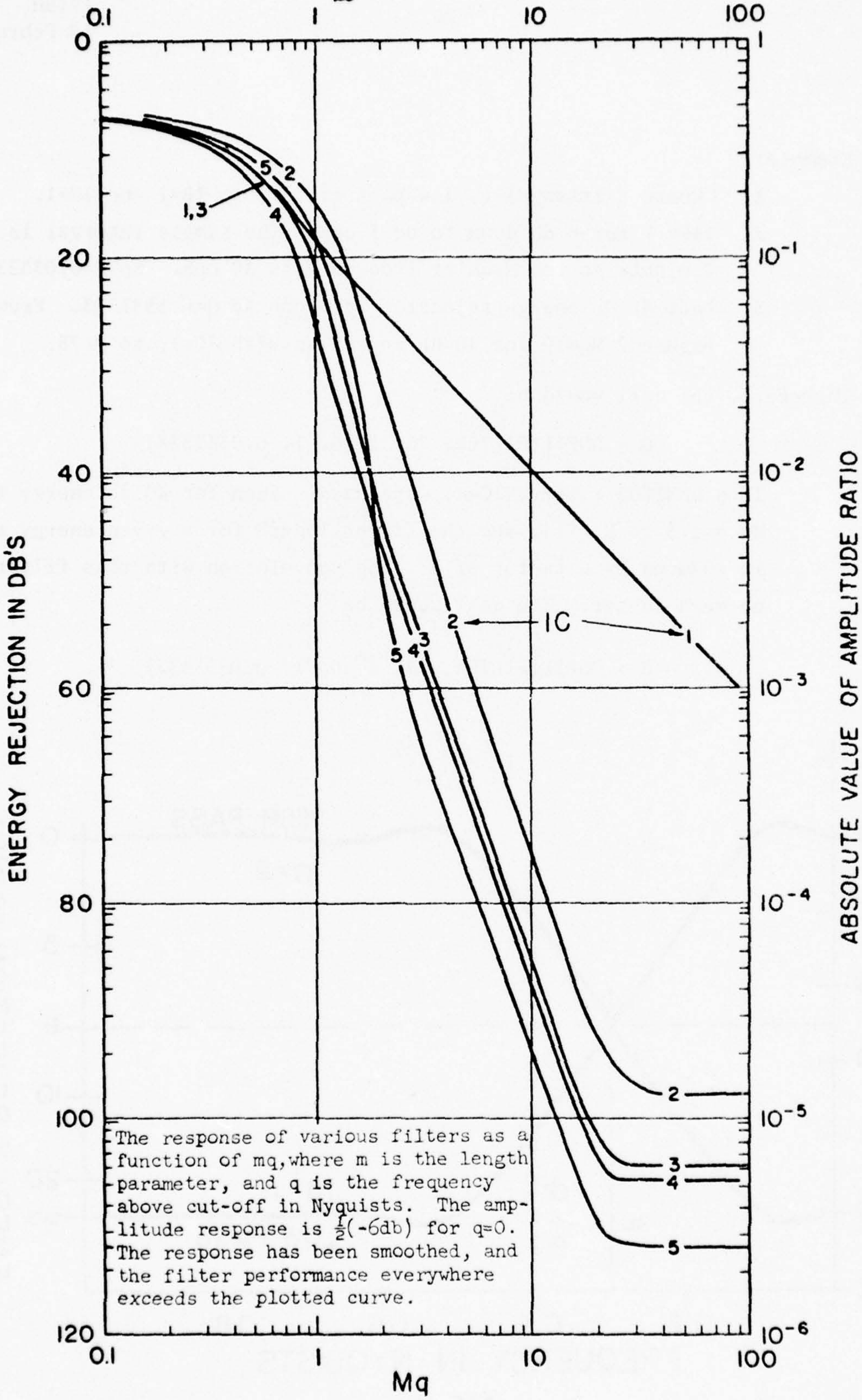


FIG. 1



ENERGY REJECTION IN DB'S
ABSOLUTE VALUE OF AMPLITUDE RATIO
Mq
0.1
1
10
100
0
20
40
60
80
100
120
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
1,3
5
2
4
2
IC
1
5
4
3
2
3
4
5
The response of various filters as a function of mq, where m is the length parameter, and q is the frequency above cut-off in Nyquists. The amplitude response is $\frac{f}{2}$(-6db) for q=0. The response has been smoothed, and the filter performance everywhere exceeds the plotted curve.

FIG. 2

NCONVL (SERIES, LOS, FILTER, LF, RESULT, LR, JALL, JADV, JXTR, IDROPZ)

Function NCONVL computes RESULT, the convolution product of SERIES and FILTER according to the parameters set (see below).

Input:

SERIES	=	array of dimension LOS.
LOS	=	dimension of series SERIES.
FILTER	=	array of filter weights of dimension LF.
LF	=	dimension of series FILTER.
LR	=	dimension of resultant convolution product RESULT, depends on LF, JXTR and IDROPZ (see below).
JALL	=	0 FILTER(1) through FILTER (LF) represent the center term and right half of a symmetric filter (as generated by GNFIL when IA = 0).
	=	1 if FILTER (1) through FILTER (LF) represents the entire series.
JADV	=	number of terms in SERIES which the series FILTER is advanced (the decimation factor). e.g., if JADV = 1, then the maximum number of lagged products is produced, when JADV = 2, every other lagged product is produced, etc.
JXTR	=	0 for normal application. For convolution in parts see further writeup in program listing.
IDROPZ	=	0 for normal application. If set = 1, then the zeroes are truncated from the ends of FIL and the resultant product may be up to 2 terms longer. For further details see writeup in program listing.

Output:

RESULT = array of convolution product series.

The function sets NCONVL = 0 if the call was successful.

= 1 if LOS or LF or LR ≤ 0

Define the effective length of FILTER to be LE

$$\text{JALL} \neq 0 \qquad \text{LE} = \text{LF}$$

$$\text{JALL} = 0 \qquad \text{LE} = 2*\text{LF}-1$$

$$\text{LR} = \text{INTEGER }[(\text{LOS}-\text{LE}+\text{JADV})/\text{JADV}]$$

Let SERIES (1) through SERIES (LOS) be the series to be convoluted, FILTER(1) through FILTER (LF) be the filter weights and RESULT (1) through RESULT(LR) be the resultant convolution product. Then,

If JALL = 0, RESULT is computed by

$$\text{RESULT(J)} = \frac{1}{\text{LF}} \sum_{I=1}^{\text{LF}} \text{SERIES (K-I+1)*FILTER(I)}, \text{ for J} = 1, 2, \ldots, \text{LR}$$

where K = LE+(J-1)*JADV.

If JALL ≠ 0, RESULT is computed by

$$\text{RESULT(J)} = \frac{1}{2\text{LF}-1} \{\text{SERIES(K)*FILTER(1)} + \sum_{I=2}^{\text{LF}} [\text{SERIES(K-I+1)}]\text{*FILTER(I)}\}$$

for J = 1, 2,...LR where K=LF+(J-1)*JADV.

Example: to lowpass and decimate a series,

SERIES(1) was sampled with ΔT = 1 min. We want to decimate to ½ hour sample interval suppressing high frequencies to prevent aliasing. Generate filter

G = GNFIL(FILTER, 15, 5, 0, 1, 0.0333333)

For the convolution product, take LOS = 2384. Then let JXTR = 0, and IDROPZ = 0. JALL = 0, since FILTER was generated by GNFIL with IA = 0. JADV = 30 to make sample interval 30 minutes instead of 1. Then LR can be found to be 80 and the call would then be

N = NCONVL(SERIES, 2384, FILTER, 16, RESULT, 80, 0, 30, 0, 0)

NCROSC(ASER,NSER,LOS,BSER,RSER,LP,FMEAN,NISHUL)

Function NCROSC calculates the Cross-Correlation, RSER, of two single component series, ASER and BSER, for lags 0 through LP. (If BSER is set to "4HNONE," RSER consists of the merged series of all Auto- and Cross-Correlations between the component series of ASER for lags 0 through LP.) ASER and BSER must have the same number of terms and RSER must fit in memory. The terms of RSER are:

$$RSER_j = \frac{1}{LOS} \sum_{i=j}^{LOS-1} ASER_i \cdot BSER_{i-j}, \quad j = 0,1,\ldots,LP.$$

Input:

ASER	=	Array of dimension NSER by LOS containing LOS terms of NSER series in merged form
NSER	=	Number of merged component series in ASER
LOS	=	Number of terms in the component series in ASER and BSER
BSER	=	Array of dimension LOS containing the single component series for cross-correlation with the single components series in ASER. If BSER is set equal to "4HNONE," the program assumes the series are stored in merged form in ASER and ignores BSER.
FMEAN	=	Array of dimension 2 containing the means of ASER and BSER respectively, which are subtracted before the correlation is computed. If NSER > 1, it is assumed that the means are removed from the component series in ASER
LP	=	The cross-correlation or lagged product is calculated for lags equal to 0 through LP
NISHUL	=	0 if the call is the last call but not the first
	=	1 if the call is the first call
	=	2 if the call is not the first or last call. This is used when series are long and the correlation is done by dividing the series into sections and calling NCROSC for each section

Output:

RSER = Array of dimension NSER * NSER * (LP + 1) which contains the cross-correlation of ASER and BSER or the Auto- and Cross-Correlations of the component series in ASER. See example below for order of merged results.

the function sets NCROSC

= 0 if the call was successful

= 1 if NSER or LOS or LP $\leq$ 0

Example:

Let ASER (1, LOS) = series 1

ASER (2, LOS) = series 2

BSER (1) = 4HNONE

FMEAN(1) = 0.

FMEAN(2) = 0.

NISHUL = 1

then the call would be

N = NCROSC (ASER, 2, LOS, BSER, RSER, FMEAN, LP, NISHUL)

and if N = 0 there were no errors. If R = RSER, then

R(1, 1) = Auto-correlation of Series 1

R(1, 2) = Cross-Correlation of Series 1 and Series 2

R(2, 1) = Cross-Correlation of Series 2 and Series 1

R(2, 2) = Auto-Correlation of Series 2

stored by lags.

PRPLOT(NAME, YMIN, YMAX, LOS, IAUTO, MOB, MLOS, NSER,SERIES)

Subroutine PRPLOT produces a printer plot of up to 5 series, each of arbitrary length. The first term of each series is plotted together on the same printer line. Shorter series will be filled with the minimum value of the plot to bring the length of all series to the length of the longest.

Input:

NAME	=	array of dimension NSER. NAME(J) contains any alphanumeric label (maximum 10 characters) of the Jth component of SERIES.
YMIN	=	array of dimension NSER. YMIN(J) contains minimum value that will be plotted for the Jth series.
YMAX	=	array of dimension NSER. YMAX(J) contains maximum value that will be plotted for the Jth series.
LOS	=	array of dimension NSER. LOS(J) contains the number of terms in the Jth series.
IAUTO	=	0, YMIN and YMAX are used as set above
	=	1, each series is examined separately and new YMIN and YMAX are selected if any value of the SERIES exceeds the values specified. Note: the limits may not be the same for each series.
	=	2, examines all series as IAUTO = 1, but plots all series at same scale. Note: "Nice" divisions are chosen automatically.
	=	3, Changes SERIES to log SERIES. Log of YMIN and YMAX will be used as specified. Note: no negatives or zeroes allowed.
	=	4, Changes SERIES to log SERIES. YMIN and YMAX will be automatically selected to nearest decade if they exceed the values specified. All series plotted on the same scale and no negatives or zeroes allowed.
MOB	=	0, any point lying outside graph limits will not be plotted.
	=	1, any point lying outside graph limits will be printed by "unwrapping" around plotting limits, i.e., SERIES(I) = MOD(SERIES(I), YMAX-YMIN)

MLOS = number of terms in longest series.

NSER = total number of series to be plotted, not to exceed 5.

SERIES = two-dimensional array of dimension (MLOS, NSER), containing all series to be plotted. Series(I,J) contains the value of the Ith term of the Jth series. SERIES(I,1), SERIES(I,2),...,SERIES(I,NSER) will be plotted together on the Ith printer line. The corresponding numerical value of the first series(J=1) will be printed along the side of the graph. Note that SERIES is the transpose of the way series are stored on GEOTAPE and used elsewhere in WHIMPER. The subroutine TPOZ(SERIES,M,N) is an in-place transpose routine which takes a M by N array and returns a N by M array. Hence a call to TPOZ will correct the problem of needing the transpose of a 2-dimensional array.

SPECTRA

This section describes programs and subprograms which create and plot power spectra of time series. There is a choice of algorithms depending on the desired result. Routines for calculating the discrete fourier transforms are:

1. Function HARM calculates the fourier coefficients by the slow method of calculating the full sums. It has the advantages of accepting series lengths which are not a power of 2, not all the coefficients need to be calculated and estimates can be made on non-integral harmonics. It has the disadvantage of being slow, extremely slow for long series (>1024).

2. Function IFTPER calculates the fourier coefficients by fast fourier techniques. The function writes the fourier coefficients on disk and/or stores the periodogram values in an array. This is used with functions ISPSMO and LGAX which smooth and plot the power density spectra.

3. Function PCROSP calculates the cross spectral matrix of an array of series by fast fourier techniques. Fuction NTRMAT then is called to produce coherence and phase.

The above routines are normalized such that the fourier coefficients returned are the amplitude of a Cosine and Sine series. The zeroth component is just the series mean. The spectral densities are then normalized so the sum of the spectra, P(f) equals the variance,

$$\sum_{f=1}^{\text{NYQUIST}} P(f)\,\Delta f = \text{VAR}.$$

Note that the sum is just over the positive frequencies.

Subroutines and functions used with the above transforms are:

1. Subroutine UNSORT takes output of HARM and creates periodogram, real and imaginary transform coefficients or amplitude and phase.

2. Subroutine TAPER applies a Cosine taper to the first and last 10% of the data in time space. It is used to reduce the effects of the gating of the data.

3. Function ISPSMO smoothes the periodogram values produced by IFTPER and corrects for constant filter response.

4. Function LGAX plots the output of ISPSMO as a log-log spectrum on either the printer or cal-comp.

5. Function NTRMAT performs a matrix transformation on the output of PCROSP to give coherence and phase as well as spectra and correct for filter response.

6. Subroutine PRCOH produces a printer plot of coherence and phase.

7. Subroutine CCCOH produces a cal-comp plot of coherence and phase with a log axis similar to axis used by LGAX to plot spectra.

TAPER (SERIES, LOS)

Subroutine TAPER applies to a cosine taper to the first and last 10 percent of the series SERIES. The tapered series is returned in the same array (altering the original series).

$$SERIES\ (j) = 0.5(1-COS(10\pi(j-1)/LOS))*SERIES\ (j)$$

where j is the distance from both ends of the series and $1 \leq j \leq LOS/10$.

NOTE this changes the resultant series so an amplitude now has 0.9 the value it previously had, so all spectral values must be correspondingly corrected:

Tapered Amplitudes = 0. 935* True Amplitude
Tapered Spectrum = 0. 875 *True Spectrum

HARM (SERIES, Z, LOS, FS, DF, NFREQ)

Function HARM calculates the fourier coeeficients, Z, of a given SERIES, where

$$SERIES_k = \sum_{j=1}^{LOS/2} Z_j \emptyset_{kj} \ , \quad k = 1, 2, \ldots, LOS$$

and

$$\emptyset_{kj} = \begin{Bmatrix} COS \\ SIN \end{Bmatrix} \left(\frac{2 \cdot \pi \cdot DF \cdot (FS+J-1)}{LOS} k\right), \quad \begin{Bmatrix} j=odd \\ j=even \end{Bmatrix}$$

where FS = 0.(1)

The terms of Z are given by

$$Z_j = \frac{2}{LOS} \sum_{k=1}^{LOS} SERIES_k \cdot \emptyset_{kj}, \quad j = 1,2,\ldots 2*NFREQ$$

If FS = 0, then

$$Z_1 = \frac{1}{LOS} \sum_{k=1}^{LOS} SERIES_k = \text{mean.}$$

Input:

SERIES - array of values to be transformed

LOS - length of array SERIES

FS - the first term of Z is at FS harmonics. (1 harmonic = 1/LOS). FS is usally zero. FS need not be in even harmonics.

DF - internal between terms of Z in harmonics. DF in usally 1. DF need not be in even harmonics.

NFREQ - number of frequencies at which the transform is calculated. NFREQ is usually LOS/2 for a complete transform.

Output:

Z - array of dimension 2*NFREQ containing the fourier coefficients.

The function HARM = 0 if the call is successful
= 1 if 2*NFREQ > LOS

1). Not all the fourier coefficients need be calculated. By setting FS $\neq$ 0, the first coefficient will be estimated at frequency FS. (not necessarily an even harmonic). The sum is only calculated NFREQ times so considerable computational time can be saved by using HARM when only a few spectral lines are desired.

UNSORT (Z, OUT, DIMZ, IO)

Subroutine UNSORT takes the results of HARM and alters it in accordance with IO.

Input:

Z = fourier coefficients produced by HARM.

DIMZ = dimension of Z = 2*NFREQ in HARM.

Output:

For IO = 1, the dimension of OUT = DIMZ

OUT = series of alternating amplitude and phase terms.

For IO = 2, the dimension of OUT = DIMZ/2

OUT = series of amplitude squared ($A^2 + B^2$) for use as periodogram.

For IO = 3, the dimension of OUT = DIMZ/2

OUT = series of log of amplitude squared.

IFTPER (SERIES, LOS, S, DT, DF, N1, IFLAG)

Function IFTPER computes the fast fourier transform and periodogram of a series SERIES of length LOS, where LOS is a power of two less than or equal to 16,384. The transform is computed at LOS/2 + 1 frequencies and its value at the Kth harmonic (1 harmonic = 1/(LOS*DT)) is

$$Z_K = \frac{2}{LOS} \sum_{j=1}^{LOS} SERIES_j * EXP(-2 \cdot \pi \cdot i \cdot (j-1)\ (K-1)/LOS)$$

except at K=0 (zero frequency) and K = LOS/2 (Nyquist frequency) where $Z_K = 1/2 \cdot Z_K$, the value of the periodogram at the Kth frequency is

$$P_K = 2 \cdot (Re(Z_I)^2 + Im(Z_I)^2)$$

except at K=0 (zero frequency) and K = LOS/2(Nyquist frequency) where $P_K = 1/2 \cdot P_K$, so the periodogram sums to the variance over the positive frequencies.
If DT≤0, then the function returns SERIES(K) = P(K).
If DT>0, then SERIES(K) = LOS•DT•P(K). (This gives units of spectral density.)

Input:

SERIES = Data to be transformed in SERIES(1),...SERIES(LOS)
NOTE: SERIES must be dimensioned at least LOS+2. The periodogram is returned in SERIES(1),...,SERIES(N1) and if DT≠0 and IFLAG(1)≠0, frequencies 0, DF = 1/(LOS•DT). 2DF,..., (LOS/2)•DF are returned in SERIES(N1+1),...,SERIES (2•N1).

LOS = number of input data points which must be a power of 2(2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384).

S = array where sines and stored by FFT subroutine.
S must be dimensioned at least LOS/4 and not used by the calling program between calls to IFTPER.

DT = time or distance between data points if periodogram is to be normalized or frequencies generated. Units of frequencies are inverse units of DT.

IFLAG input parameters; (IFLAG dimensioned at least 9)

IFLAG (1) nonzero to generate frequencies if DT > 0.

IFLAG (2) > 0 to store transform in next record of tape or disk file IFILE = IFLAG(2). SERIES is LOS+2 long with the real part of the transform in words 2K-1 and the imaginary part in words 2K.

IFLAG (3) nonzero to compute and print mean and variance of data before transforming.

IFLAG (4) nonzero to compute and print mean and variance of data from transform as a check that transform was done correctly.

IFLAG (5) nonzero to compute and remove mean of data before transforming.

IFLAG (6) ≥ 0 to print P input data points.
If IFLAG (6) is between LOS/2 and LOS, the first P=IFLAG(6) points are printed.
If IFLAG (6) ≥ LOS all P=LOS points are printed.
If IFLAG (6) ≤ LOS/2, the first P/2=IFLAG(6) and last P/2 points are printed.

IFLAG (7) ≥ 0 to print P transform values as defined for IFLAG (6) except that LOS+2 is used instead of LOS. Remember that odd indexed points are real part of transform and even indexed are imaginary.

IFLAG (8) ≥ 0 to print P periodogram values as defined for IFLAG(6) except that N1 is used instead of LOS and IFLAG(8) instead of IFLAG(6).

IFLAG (9) ≥ 0 to print P frequencies as defined in IFLAG(8).
This is ignored if IFLAG(1) = 0 or DT ≤ 0.

Output:

SERIES = See input

DF = harmonic value = 1/(LOS·DT). Periodogram values are returned frequencies 0, DF,...,(LOS/2)·DF where (LOS/2)DF=1/(2·DT)=Nyquist frequency.

N1 = number of periodogram values returned = LOS/2+1.

The function IFTPER = 0 if call is successful.

= -1 if FFT subroutines are told to compute the sines unnecessarily

= 1 if FFT subroutine has error

= 2 if LOS is not a power of 2.

ISPSMO (SERIES, N1, CF, DF, AVE, NAVG, NDIM, NPPL, JFLAG)

Function ISPSMO does normalization of (spectral) estimates in SERIES and smooths over equal intervals of frequency or log-frequency. Smoothing over linear frequency intervals can be rectangular or triangular, with or without decimation. Smoothing over log-frequency intervals is rectangular with decimation.

Input:

SERIES = data to be smoothed in SERIES(1),...,SERIES(N1). Smoothed values are returned in SERIES(1),...,SERIES(NPPL) If DF $\leq$ 0 the user must put frequencies corresponding to SERIES(1),...SERIES(N1) in SERIES(N1+1),...,SERIES(2·N1). If DF > 0, frequencies 0, DF, ...(N1-1)·DF are automatically put in SERIES(N1+1),...SERIES(2·N1). Then frequencies corresponding to the smoothed values replace SERIES(NPPL+1),... SERIES(2·NPPL).

N1 = number of input SERIES values in periodogram.

CF = normalization constant by which smoothed values are multiplied. Use CF = 1. if desired normalization has been done, e.g., by IFTPER. If CF $\leq$ 0, ISPSMO uses CF = 1. Used to remove constant prefilter response.

DF = >0 if frequencies are to be generated as described above.

AVE = frequency band in log units used in boxcar averaging over log-frequency intervals when JFLAG(2) = 0. Must have AVE > 0 if JFLAG(2) = 0.

NAVG = array dimensioned at least NDIM in calling program. The number of points averaged to obtain SERIES(I) is returned in NAVG(I), I = 1,...NPPL.

NDIM = maximum number of points returned. High frequency values are not returned if the smoothing specified results in too many points.

JFLAG input parameters (JFLAG dimensioned at least 6)

JFLAG (1) $\leq$ 0 includes all SERIES values in averaging.

$>$ 0 just normalizes, does not average, the first JFLAG(1) values.

JFLAG (2) $<$ 0 for simple moving average smoothing with no decimation. Each smoothed value is a weighted average (see JFLAG(3)) of $2 \cdot |JFLAG(2)|+1$ input values corresponding to each index value from $JFLAG(1) + 1+|JFLAG(2)|$ to $N1 - |JFLAG(2)|$. Use $JFLAG(2) < 0$ only when N1 is small. If $JFLAG(3) \neq 0$ and $|JFLAG(2)|$ is odd, $|JFLAG(2)| + 1$ is used instead of $|JFLAG(2)|$.

= 0 for boxcar smoothing over equal log frequency intervals. Points are returned only at frequencies $>$ 0 unless $JFLAG(1) \geq NDIM$.

$>$ 0 for smoothing over intervals of $2 \cdot JFLAG(2) + 1$ points with decimation (only every $[2 \cdot JFLAG(2)+1]$th point is kept).

JFLAG(3) = 0 for rectangular (boxcar) smoothing with weights $1./(2|JFLAG(2)|+1)$.

$\neq$ 0 for triangular smoothing with weight i frequencies away from center frequency $(|JFLAG(2)|+1-i)/(|JFLAG(2)|+1)^2$ when $JFLAG(2) \neq 0$.

JFLAG(4) $\neq$ 0 to print SERIES before smoothing

JFLAG(5) $\neq$ 0 to print SERIES after smoothing

JFLAG(6) $\neq$ 0 to print NAVG

Output:

SERIES = See Input

NPPL = number of smoothed points

The function ISPSMO = 0 if call is successful

= 1 if $N1 \leq 0$ or $NDIM \leq 0$.

= 2 if ISPSMO does no smoothing because JFLAG(2)=0 and either $AVE \leq 0$, or there are too many frequencies

= 3 if ISPSMO set $NPPL > N1$.

FUNCTION LGAX (SERIES, NPPL, LABELS, KFLAG)

Function LGAX produces a Cal-Comp or printer plot of points in array SERIES and connects the points by straight lines on the Cal-Comp plotter. The function can be used to draw a log-log grid and/or plot points. LGAX chooses proper grid limits from the values to be plotted. If the user wishes to set his own limits then call LGAX twice, first with SERIES(1)=YMIN, SERIES(2)=YMAX, SERIES(3)=XMIN, SERIES(4)=XMAX, NPPL=2, IFLAG(6) < 0 and IFLAG(7)≠0 to draw the grid, and second with SERIES equal the values to be plotted and IFLAG(6) > 0. Finished plots fit on 8.5 x 11 paper with 1 inch/cycle if the X-axis is 4 to 6 cycles or the Y-axis 5 to 9 cycles and 2 inches/cycle if the X-axis is 3 cycles and the Y-axis 3 to 4 cycles. Plot requires routines on permanent file GRAFIX.

Input:

SERIES = series to be plotted with Y-values in SERIES(1),...SERIES(NPPL) and X-values in SERIES(NPPL+1),...,SERIES(2*NPPL).

NPPL = number of points to be plotted.

LABELS = array of labels as described below. Must be dimensioned at least 5 for Cal-Comp.

KFLAG input parameters, KFLAG dimensioned at least 7.

KFLAG (1) <0 to terminate plot or change plot device by calling EXITPL after this plot. User can also terminate plot by calling EXITPL in mainline program after call to LGAX.

= 0 normal call except to for first or last plot on a device

> 0 to initialize plot and select plot device

if = 1, LGAX calls STCCON for a Cal-Comp plot

if > 2, LGAX calls PRNTON for a printer plot

KFLAG (2) ≤ 0 for no identifying label at bottom of plot, and i=0.

> 0 then KFLAG(2) is the number of characters (N≤80) in identifying label which is stored in LABELS(1)...LABELS(i) where i = (N+9)/10 (integer arithmetic)

KFLAG (3) $\leq$ 0 for no X-axis label or annotation, and j = 0

$>$ 0, then KFLAG(3) is the number of characters (NX) in X-axis label stored in LABELS (i+1)...,LABELS (i+j) where j = (NX + 9)/10. For X-axis annotation (grid values) but no label use NX=1 and LABELS (i+1) blank.

KFLAG (4) $\leq$ 0 for no Y-axis label or annotation.

$>$ 0, then KFLAG(4) is the number of characters (NY) in Y-axis label stored in LABELS (i+j+1),...,LABELS (i+j+k) where k = (NY+9)/10. For annotation but no label, set NY=1 and LABELS (i+j+1)blank.

KFLAG (5) = 0 to plot input values in SERIES.

$\neq$ 0 to replace SERIES by $SERIES_i = SERIES_i \cdot FREQ^{KFLAG(5)}$ and plot the modified series (spectrum times power of frequency).

KFLAG (6) $<$ 0 to draw grid only.

= 0 to draw grid and plot points

$>$ 0 to plot points only

If KFLAG(6) $\geq$ 0, all X and Y values outside the plot limits are plotted at the limit and the values in SERIES replaced by the plotted values.

KFLAG (7) = 0 to advance to next graph after plotting.

$\neq$ 0 do not advance (use if more points are to be plotted on same grid with another call to LGAX).

The function LGAS = 0 if the call is successful.

= 1 if NPPL $\leq$ 0.

Example:

Read series from permanent file, create periodogram, smooth and plot on log-log scales with labels.

```
SPECTRA, CM100000.
ACCOUNT (----)
ATTACH (TAPE1, BOTTEMP, ID = STRAITS)
ATTACH (WHIMPER, ID = IRISH)
ATTACH (GRAFIX)
DISPOSE (TAPE99,*CC)
LDSET (LIB = WHIMPER/GRAFIX)
LGO.
EOR.
      PROGRAM SPECTRA (INPUT, OUTPUT, TAPE1)
      DIMENSION SERIES(2050), S(512), IFLAG(9), NAVG(100), JFLAG(6), LABELS(6),KFLAG(7)
      DATA IFLAG /1,0,1,1,1,0,0,0,0/
      DATA JFLAG /0,0,0,0,1,1,/
      DATA KFLAG /1,20,3,21,0,0,0/
      DATA LABELS /20H STRAITS BOTTOM TEMP, 3HCPH,21HDEG C SQUARED PER CPH/
      LOS = 2048
      READ(1) (SERIES(I), I = 1, LOS)
      CALL TAPER (SERIES,LOS)
      I = IFTPER (SERIES,LOS,S,0.5,DF,N1,IFLAG)
      J = ISPSMO (SERIES,N1,0.81,DF,0.02,NAVG,100,NPPL,JFLAG)
      K = LGAX (SERIES, NPPL,LABELS,KFLAG)
      CALL EXITPL
      STOP
      END
EOF
```

PCROSP (SERIES, NSER,LOS,HFS,HFE,M,SP,SCRATCH,CF,KTAPER,IE,NEWK)

Function PCROSP is used to compute the cross power spectra of all pairs of component series stored in the array SERIES dimensioned NSER by LOS. The series may be divided into sections of length LOS where LOS is a power of two. Then PCROSP is called for each section. The mean of each component of SERIES is subtracted from each term. If KTAPER is non-zero a cosine taper is applied to the first and last 10% of SERIES and the fast fourier transform of each component series is computed. The auto spectra, and cross sprectra are computed.

Designate the terms of an arbitrary section of any two component series of SERIES by $u_1,\ldots u_{LOS}$ and $v_1,\ldots v_{LOS}$. The fourier transforms are

$$U_j = \frac{2\varepsilon}{LOS} \sum_{t=1}^{LOS} u_t \bullet EXP(2\pi ij(t-1)/LOS)$$

$$V_j = \frac{2\varepsilon}{LOS} \sum_{t=1}^{LOS} v_t \bullet EXP(2\pi ij(t-1)/LOS)$$

where $j=0, 1, \ldots, \frac{1}{2}$ LOS and $\varepsilon=1$ except for $j=0$ and $j=\frac{1}{2}LOS$ where $\varepsilon=\frac{1}{2}$ the corresponding cospectrum (C) and quadrature spectrum (Q) are the real and imaginary parts of the $U_j V_j$ product averaged over all sections.

$$S_j(uv) = C_j(uv) + iQ_j(uv)$$

$$= \frac{1}{NS} \sum_{n=1}^{NS} (U_{n,j} \bullet V_{n,j})$$

where NS in the number of sections the series is divided into.

Neighboring frequencies may be combined to form triangular averages. (M=1) frequencies on both sides are combined with S_j in the weighted average. M must be a power of 2. For $M \geq 2$,

$$\overline{S_j} = S_j + \sum_{n=1}^{M-1} \frac{M-n}{M} [S_{j-n} + S_{j+n}],\ j = HFS, HFS+M, \ldots HFE$$

where $0 \leq HFS \leq HFE \leq \frac{1}{2}LOS$. $S_j = 0$ for $j \leq 0$ and $j > \frac{1}{2}LOS$. The frequency interval now becomes $\Delta F = M/(LOS \bullet DT)$ where DT is the sample interval.

The resultant spectra are stored at SP(I,J,K). This matrix can be manipulated by NTRMAT to yield phase and coherence. For the form of SP see NTRMAT. Example of use follows NTRMAT.

Input:

SERIES - array of componenet series of dimensions NSER by LOS.

NSER - number of component series in array SERIES.

LOS - number of values in each component series. Must be a power of 2.

HFS - harmonic of first spectral estimate

HFE - harmonic of last spectral estimate. Note that $HFE \leq LOS/2+1$.

M - the dimension of array CF. M is the amount of decimation in frequency, i.e., (M-1) harmonics on each side of the spectral value are combined with the central frequency to form triangular averages. M must be a power of 2. New ΔF = M*old ΔF.

SCRATCH - working array of dimension LOS. (Dimensioned in main program.)

CF - array of dimension (M-1), used to store weights for triangular averages.

KTAPER = 0 to apply no taper.
= 1 to apply cosine taper to first and last 10% of data series.

IE = 0 if this is not the first section and not the last section.
= 1 if this is the first section and not the last section.
= 2 if this is not the first section and is the last section.
= 3 if this is the first section and the last section.

NEWK - is used when different sets of consecutive harmonics are desired. PCROSP should be called twice per section with NEWK = 0 the first call and NEWK = 1 on the second. When NEWK•NE•0 the FFT is not computed. NSER and LOS must be the same as for the previous call but M may be changed.

Output:

SP - a 3-dimensional array of cross spectral values of dimension NSER by NSER by NFREQ, NFREQ = (HFE - HFS) + 1. SP(I,I,K) is the power spectrum of component series I. SP(I,J,K) where I>J is the co-spectra and SP(J,I,K) the quadrature spectra.

The function returns 0 if the call was successful and non-zero otherwise.

NTRMAT (SP, NSER,NFREQ,KOUT,TSP AMPL,PHASE)

Function NTRMAT transforms the spectral array SP created by PCROSP into other forms and corrects for instrumental response. Take a case where NSER = 2. Then SP is dimensioned 2 by 2 by NFREQ and at each frequency the 2 by 2 array consts of

$$\begin{pmatrix} S_{11} & C_{12} \\ Q_{12} & S_{22} \end{pmatrix} \qquad \text{KOUT} = 1$$

Other output forms of TSP are:

$$\begin{pmatrix} S_{11} & R^2_{12} \\ \phi_{12} & S_{22} \end{pmatrix} \qquad \text{KOUT} = 2$$

$$\begin{pmatrix} \log S_{11} & R^2_{12} \\ \phi_{12} & \log S_{22} \end{pmatrix} \qquad \text{KOUT} = 3$$

$$\begin{pmatrix} S_{11} & R_{12}\text{Cos } \phi_{12} \\ R_{12}\text{SIN } \phi_{12} & S_{22} \end{pmatrix} \qquad \text{KOUT} = 4$$

$$\begin{pmatrix} 20 + \log S_{11} & R_{12}\text{COS } \phi_{12} \\ R_{12} \text{ SIN } \phi_{12} & 20 + \log S_{22} \end{pmatrix} \qquad \text{KOUT} = 5$$

where $R^2_{12} = (C_{12}{}^2 + Q_{12}{}^2)/S_{11}S_{22}$ and $\phi_{12}= \text{ARCTAN}(Q_{12}/C_{12})$ are the squared coherence and phase spectra. ϕ_{12} is the phase lead of record 2 relative record 1 in degrees, $-180 < \phi \leq 180$.

AMPL and PHASE are arrays of dimension NFREQ giving the response function of some filter to be removed from the spectra. If $S_{ii}(f)$ is the uncorrected power spectrum and $\phi_{ij}(f)$ is the uncorrected phase, then the corrected values stored in SP are:

$$\text{TSP}_{ii}(f_k) = \text{SP}_{ii}(f_k)/\text{AMPL}_{ik}$$
$$\text{New } \phi_{ij}(f_k) = \text{old } \phi_{ij}(f_k) - (\text{PHASE}_{ik} - \text{PHASE}_{jk})$$

and R_{ij} is unchanged.

Input:

SP = Spectral array of dimension NSER by NSER by NFREQ as produced by PCROSP.

SP(i,i,k) = power spectrum of series i

SP(j,i,k) = co-spectrum for $i < j$

SP(i,j,k) = quadrature spectrum

SP(j,j,k) = power spectrum of series j.

NSER = number of component series in SP

NFREQ = number of frequencies in SP.

KOUT = 1 for normalized version of SP. Just normalizes by AMPL of respective series.

$TSP_{ij} = SP_{ij}/(AMPL_i \bullet AMPL_j)$ at each frequency.

= 2 for squared coherence, phase and power spectra.

= 3 for squared coherence, phase and log of power spectra.

AMPL = array of amplitude response of dimension NSER by NFREQ.

PHASE = array of phase response of dimension NSER by NFREQ. (in degrees).

Outputs:

TSP = Spectral array of NSER x NSER x NFREQ depending on the value of KOUT.

AMPL and PHASE are used to correct for the response of a sensor or prefilter on the data. If no correction is needed, then set AMPL(1,1) = 4H NONE and PHASE(1,1) = 4H NONE and no correction is applied.

SUBROUTINE PRCOH (ARRAY, NSER, LOS, NAME)

PRCOH produces a printer plot of coherency and phase spectra. As many a 3 coherency-phase spectra pairs may be plotted on the same graph. Coherency will be plotted on a scale from 0 to 1; any other value will produce error message on plot, but system will not dump. Phase must be in degrees and will be plotted from -180° to +180°; any other value will be plotted by "wrapping" around 180°.

Inputs:

ARRAY = two-dimension array of dimension (LOS, NSER) containing all series to be plotted. ARRAY(I,J) contains the Ith term of the coherency spectra (J=odd) and phase spectra (J=even). ARRAY(I,1) ARRAY(I,3), ARRAY(I,5), (coherency spectra) and ARRAY(I,2), ARRAY(I,4), ARRAY(I,6) (phase spectra) will be plotted together on the Ith printer line.

NSER = total number of series in ARRAY, must be either 2,4, or 6.

LOS = number of terms in each series; all series must be same length.

NAME = array of dimension NSER/2. NAME(J) contains any alpha-numeric label (maximum 10 characters) of Jth coherency-phase spectra pair.

SUBROUTINE CCCOH(SERIES, NPPL, LABELS,LFLAG)

CCCOH produces a cal-comp or printer plot of coherence squared and phase spectra as a function of log frequency. Coherency will be plotted on a scale from 0 to 1 and phase from -180 to 180 degrees.

Input:

SERIES - array of dimension 3 times NPPL containing in order, coherence, phase and frequency series in joined form, each series NPPL long.

NPPL - number of frequency estimates at which the coherence and phase will be plotted. The dimension of each of the component series in array SERIES.

LABELS - array containing the frequency axis label and plot label. See LFLAG(2) and LFLAG(3) for dimension of LABELS.

LFLAG input parameters: LFLAG dimensioned at least 4.

LFLAG (1) = 0 for normal call
= 1 for cal-comp plot initialization
= 2 for printer plot initialization

LFLAG (2) = number of characters (≤80) in the plot label which is stored in LABELS(1) through LABELS(i) where i = (LFLAG(2)+9)/10 (integer arithmetic).

LFLAG (3) = number of characters in x-axis label or annotation which is stored in LABELS (i+1) through LABELS (i+j) where j = (LFLAG(3)+9)/10 (integer arithmetic)

LFLAG (4) < 0, to draw axis only
= 0, to draw axis and plot
> 0, to plot point only.

TIDES

The tide programs are based on the independent routines developed at IGPP/UCSD for tide analysis with the addition of a few locally created subroutines. These routines analyze a time series for its tidal content.

The functions and subroutines are:

TIDPOT	-	generates the tide potential from astronomical constants.
TIHAR	-	predicts the tide published tide constants (harmonic method)
TIDWTS	-	finds the convolution weights which fit reference to observed in least squares sense.
TIDADM	-	transforms the weights to get the amplitude and phase of the principal diurnal and semidiurnal lines.
TIDRSP	-	makes a noise-free prediction by convolving the response weights with the reference series.
TIDSPC	-	fourier transform of observed series at tidal frequencies
TIDFIL	-	generates various filters to select against tidal oscillations.

Time is measured in total Greenwich hours counted from the turn of the century (GEOTIME). Table 1 enables one to calculate these times. For example, 1615 local time in Honolulu on 25 Oct 1975 would be

1975	-	657432.
Oct	-	6552.
25	-	576.
local: 1615		16.25
time zone + 10:	-	10.
GEOTIME	-	664586.25

Further theoretical description can be found in Munk and Cartwright, 1966: Tidal Spectroscopy and Prediction, Phil, Trans. Roy. Soc. A259. A flow chart of suggested technique for using the program can be found in Cartwright, Munk and Zetler, 1969, Pelagic tidal measurements, EOS, 50, 472-477.

GEOTIME

GREENWICH HOURS

From Start Century To Start Year

Year	Hours	Year	Hours
1900	0	1950	438288
1901	8760	1951	447048
1902	17520	1952*	455808
1903	26280	1953	464592
1904*	35040	1954	473352
1905	43824	1955	482112
1906	52584	1956*	490872
1907	61344	1957	499656
1908*	70104	1958	508416
1909	78888	1959	517176
1910	87648	1960*	525936
1911	96408	1961	534720
1912*	105168	1962	543480
1913	113952	1963	552240
1914	122712	1964*	561000
1915	131472	1965	569784
1916*	140232	1966	578544
1917	149016	1967	587304
1918	157776	1968*	596064
1919	166536	1969	604848
1920*	175296	1970	613608
1921	184080	1971	622368
1922	192840	1972*	631128
1923	204600	1973	639912
1924*	210360	1974	648672
1925	219144	1975	657432
1926	227904	1976*	666192
1927	236664	1977	674976
1928*	245424	1978	683736
1929	254208	1979	692496
1930	262968	1980*	701256
1931	271728	1981	710040
1932*	280488	1982	718800
1933	289272	1983	727560
1934	298032	1984*	736320
1935	306792	1985	745104
1936*	315552	1986	753864
1937	324336	1987	762624
1938	333096	1988*	771384
1939	341856	1989	780168
1940*	350616	1990	788928
1941	359400	1991	797688
1942	368160	1992*	806448
1943	376920	1993	815232
1944*	385680	1994	823992
1945	394464	1995	832752
1946	403224	1996*	841512
1947	411984	1997	850296
1948*	420744	1998	859056
1949	429528	1999	867816

* Leap Year

From Start Year To Start Month

	Not Leap Year	*Leap Year
Jan	0	0
Feb	744	744
Mar	1416	1440
Apr	2160	2184
May	2880	2904
Jun	3624	3648
Jul	4344	4368
Aug	5088	5112
Sep	5832	5856
Oct	6552	6576
Nov	7296	7320
Dec	8016	8040

From Start Month To Start Day

Day	Hours
1	0
2	24
3	48
4	72
5	96
6	120
7	144
8	168
9	192
10	216
11	240
12	264
13	288
14	312
15	336
16	360
17	384
18	408
19	432
20	456
21	480
22	504
23	528
24	552
25	576
26	600
27	624
28	648
29	672
30	696
31	720

Example:

	1961	534720
	July	4344
	11	240
0930	PST	9.5
	GMT=PST+	8
		539321.5 hours

TIDPOT (SERIES, LGAMMA, MORDER, NDEGRE, NUMGMN, LF, COLAT, GELONG, ST, DT, ET, INITIAL)

Function TIDPOT generates functions related to the tide potentials. The functions are computed directly from the known orbital constants, (See Munk and Cartwright 1966, Tidal Spectroscopy and Prediction, Phil, Trans. Roy. Soc. A259, 533-581) See Note 1.

Inputs:

LGAMMA	=	1 for Moon's gravitational (MG) potential (see note 2)
	=	2 for Sun's gravitational (SG) potential.
	=	3 for total gravitational (G) potential (1 + 2)
	=	4 for Sun's radiational (R) potential See Note 3.
MORDER	=	Refers to spherical harmonic order (m)
NDEGRE	=	Refers to spherical harmonic degree (n)
NUMGMN	=	number of Gamma, m,n triplets desired which is also the number of merged complex series in output SERIES.
LF	=	0 for generation of $\gamma G_n^m(t)$ with COLAT and GELONG IGNORED.
	=	1 for generation of $\gamma F_n^m(t)$ which is the m,n contribution to the γ equilibrium tide observed at COLATitiude = θ and Greenwich East LONGitude = ϕ [in degrees]
COLAT	=	colatitude, ignored if LF = 0
GELONG	=	Greenwich east longitude, ignored if LF = 0
ST	=	Start time in Greenwich hours
DT	=	Sample interval in Greenwich hours
ET	=	End time in Greenwich hours.
INITIAL	=	Set non-zero on initial call. In further calls may be zero if the only parameters changed

Output:

SERIES is the merged, complex series (containing a_n^m, b_n^m for each γ,mn triplet and time) $C_n^m(t)$ or $F_n^m(t)$ depending if LF = 0 or 1 respectively.

$$C_n^m(t) = a_n^m(t) + i\, b_n^m(t) \text{ (see summary of formulae)}$$

is the complex, time varying amplitude of the spherical harmonic of order m, degree n such that

$$ {}_{\gamma}f_n^{\,m}(\theta,\phi,t) = {}_{\gamma}a_n^m(t)\,U_n^{\,m}(\theta,\phi) + {}_{\gamma}b_n^m(t)\;V_n^{\,m}(\theta,\phi) $$

$$ = \mathrm{Re}\,\{ {}_{\gamma}F_n^{\,m}(\theta,\phi,t) \} $$

$$ = \mathrm{Re}\,\{ {}_{\gamma}C_n^{\,m}(t)^{*}\,{}_{\gamma}Y_n^{\,m}(\theta,\phi) \} $$

is the m,n contribution to the γ equilibrim tide observed at colatitude θ, and east longitude ϕ in degrees.

Here

$$ Y_n^{\,m}(\theta,\phi) = U_n^{\,m}(\theta,\phi) + iV_n^m(\theta,\phi) $$
$$ = (-1)^m \left[\frac{2n+1}{4\pi}\right]^{\frac{1}{2}} \left[\frac{(n-m)!}{(n+m)!}\right]^{\frac{1}{2}} P_n^m(\cos\theta)e^{im\phi} $$

where

$$ P_n^{\,m}(\mu) = \frac{(1-\mu^2)^{\frac{1}{2}m}}{2^n\,n!}\,\frac{\partial^{\,m+n}}{\partial\mu^{m+n}}\left[(\mu^2-1)^n\right], \quad n \geqslant 0,\ |m|<n $$

in the associated Legendra function

$P_0^0 = 1$

$P_1^0 = \cos\theta$ $\quad P_1^1 = \sin\theta$

$P_2^0 = {}^3\!/_2\cos^2\theta - {}^1\!/_2$ $\quad P_2^1 = 3\sin\theta\cos\theta$ $\quad P_2^2 = 3\sin^2\theta$

$P_3^0 = {}^5\!/_2\cos^3\theta - {}^3\!/_2\cos\theta$ $\quad P_3^1 = {}^3\!/_2\sin\theta(5\cos^2\theta - 1)$ $\quad P_3^2 = 15\sin^2\theta\cos\theta$ $\quad P_3^3 = 15\sin^3\theta$

a summary of formulae and comparison with the DOODSON method follow.

SUMMARY OF FORMULAE

	TIDPOT	EQUILIBRIUM TIDE
Gravitational, MG, SG, G	$a_n^0 = K_n(\bar{R}/R)^{n+1}\frac{4\pi}{2n+1}U_n^0(\cos Z, L);\ b_n^0 = 0$	$f = a\frac{M}{M_\oplus}\xi\{(1+\xi^2-2\xi\cos\alpha)^{-\frac{1}{2}} - (1+\xi\cos\alpha)\}$ [for ☾]
	$\begin{Bmatrix} a \\ b \end{Bmatrix}_n^m = 2K_n(\bar{R}/R)^{n+1}\frac{4\pi}{2n+1}\begin{Bmatrix} U \\ V \end{Bmatrix}_n^m(\cos Z, L)$	$= a\frac{M}{M_\oplus}\xi\{-\frac{1}{2}\xi^2+\frac{3}{8}\gamma^2+\frac{5}{16}\gamma^3+\frac{35}{128}\gamma^4+..\},\ \gamma = 2\xi\cos\alpha-\xi^2$ [for ⊙]
	$n = 2,3,4,\ ..\qquad 1 \le m \le n.$	$a = 6.378 \times 10^8$ cm, $\qquad \xi = \bar{\xi}\,(\bar{R}/R)$

☾: MG; $K_n = 7.845 \times 10^6\,\bar{\xi}^{n+1}$, $\bar{\xi} = 1.65925 \times 10^{-2}$, $M/M_\oplus = 1.2300 \times 10^{-2}$; $\bar{R}/R$, cos Z, L; cos α from § 4.

⊙: SG; $K_n = 2.124 \times 10^{14}\bar{\xi}^{n+1}$, $\bar{\xi} = 4.26345 \times 10^{-5}$, $M/M_\oplus = 3.3296 \times 10^5$; $\bar{R}/R$, cos Z, L; cos α from § 5.

☾,⊙: G; Sum above two.

	TIDPOT	EQUILIBRIUM TIDE
Radiational, R	$a_n^0 = \varkappa_n(\bar{R}/R)^2\frac{4\pi}{2n+1}U_n^0(\cos Z, L);\ b_n^0 = 0$	$f = (\cos\alpha - \frac{1}{4})(\bar{R}/R)^2,\qquad 0 \le \alpha \le \frac{1}{2}\pi$
	$\begin{Bmatrix} a \\ b \end{Bmatrix}_n^m = 2\varkappa_n(\bar{R}/R)^2\frac{4\pi}{2n+1}\begin{Bmatrix} U \\ V \end{Bmatrix}_n^m(\cos Z, L)$	$f = \quad(\ - \frac{1}{4})(\bar{R}/R)^2,\qquad \frac{1}{2}\pi \le \alpha \le \pi$
	$n = 1,2,4,6,\ ..\qquad 1 \le m \le n.$	

⊙: $c = 4$; $\varkappa_1 = \frac{1}{2}$, $\varkappa_n = \frac{2n+1}{2}\left[\frac{(1)(-1)\ ..\ (3-n)}{(2)(4)\ ..\ (2+n)}\right]$; $\bar{R}/R$, cos Z, L, cos α from § 5.

$n = 2,4,6,\ ..$

$$\begin{Bmatrix} U \\ V \end{Bmatrix}_n^m(x,y) = (-1)^m\left[\frac{2n+1}{4\pi}\right]^{\frac{1}{2}}\left[\frac{(n-m)!}{(n+m)!}\right]^{\frac{1}{2}}P_n^m(x)\begin{Bmatrix} \cos \\ \sin \end{Bmatrix}(my)$$

Comparison with Doodson Method

	${}_G C_n^m$ Constituents		Doodson Constituents		
	H_{cm}	G°	Number (lunar)	Coefficient	× 69.6 cm
Q_1	4.89	179.92	1 -2 0 1 0 0	.07216	5.02
			1 -1 0	.01360	0.95
Ø1	25.97	180.08	1 -1 0 0 0 0	.37689	26.23
			0 -1 0	.07105	4.95
P_1	12.22	180.09	1 1 -2 0 0 0	.17554	12.22
K_1	36.86	180.02	1 1 0 0 0 0	-.16817 11.70 -.36233 25.22	36.92
			0 1 0	-.07182	5.00
N_2	12.08	1.56	2 -1 0 1 0 0	.17387	12.10
M_2	63.08	-0.03	2 0 0 0 0 0	.90812	63.21
			0 -1 0	.03386	2.36
S_2	32.62	-0.29	2 2 -2 0 0 0	.42286	29.43
K_2	8.28	-0.37	2 2 0 0 0 0	.07858 5.47 .03648 2.38	7.85
			0 1 0	.03423	2.38

The convention $(-1)^m$ leads to a phase angle (Greenwich epoch) $G = m\pi$. The constituents so obtained are roughly $69.6 \times |\text{Doodson coefficients}|$, where $69.6 = \frac{1}{2}\sqrt{\frac{24\pi}{5}}\,\frac{M_☾}{M_\oplus}\,\overline{\xi^3}a$. Slight discrepancies result because Doodson coefficients add $C_2^2 + C_3^2$; we do not. Doodson coefficients at nodal separation are given below the principal coefficients. For K_1 and K_2 there are two contributions at precisely the same frequency. To correct to solar Doodson numbers (as in TIDHAR), add 0 -1 1 0 0 0 for diurnals, 0 -2 2 0 0 0 for semidiurnals.

Example:

The Parameters

LGAMMA = (4, 3, 1, 4, 3, 2)
MORDER = (1, 1, 1, 2, 2, 2)
NDEGRE = (2, 2, 2, 2, 2, 2)
NUMGMN = 6
LF = 0
ST = 613608
DT = 0.5
ET = 614279.5
INISHL = 1

Create the six merged complex series.

$$ {}_{R}C_2^1 \, , \; {}_{G}C_2^1 \, , \; {}_{MG}C_2^1 \, , \; {}_{R}C_2^2 \, , \; {}_{G}C_2^2 \, , \; {}_{SG}C_2^2 $$

SERIES contains 12 merged series consisting of

$\mathrm{Re}\{{}_{R}C_2^1\}$, In $\{{}_{R}C_2^1\}$, Re $\{{}_{G}C_2^1\}$, etc.

starting at 0000 UT 1 Jan 1970 by 0.5 hour intervals through 2330 UT 28 Jan 1970.

NOTE 1 - The following corrections have been made to Munk and Cartwright 1966.

1. $\omega_\Theta = 0.4093198 - 0.0002271 \cdot T$
2. Equation A17, replace T^3 by T^2
3. Equation A18, add term $-3me_\Theta \sin(h_\Theta - p_\Theta)$
4. Equation A19, add term $- {}^3/_2 m^2 e_\Theta \cos(h_\Theta - p_\Theta)$ inside[]
5. m = 0.074801 (sidereal month/sidereal year)

NOTE 2 - Gravitation terms refer to cm of equilibrium height. To obtain equilibrium pressure in dynes cm^{-2}, multiply by $\rho g = (1.025)(981.0) = 1005.5$ for ocean tides near the surface, and by $(1.037)(981.) = 1017.3$ for ocean tides at 4 km depth. Multiply by $(0.00124)(981.) = 1.2164$ for atmospheric tide. To obtain variations in gravity, multiply by $-(1 + \frac{2}{n}h - \frac{n+1}{n}k)\ \frac{ng}{a} =$ $-(1 + 0.59 - {}^{3}/_{2} \cdot 0.29)(2 \times 980.7/6.371 \times 10^{8}) = 3.556$ μgals for $n=2$.

NOTE 3 - Radiational terms are in fractions of the solar constant, $S = 1.946$ cal cm^{-2} min^{-1}. All radiational coefficients for $n = 3, 5, 7, \ldots$ are zero to the present order. Following a suggestion by Gordon Goves, $(\bar{R}_{\Theta}/R)$ has been replaced by $(\bar{R}_{\Theta}/R)^{2}$ in the definition of the radiational function.

TIDHAR (SERIES, LTERMS, NGIVEN, NUSED, INFER, HMIN, KAPPA, ELONG, KOMPLX, ISUM, NOCARD, ALLIST, ET, ST, DT, BEGIN)

Function TIDHAR generates the tide by the harmonic method from published harmonic constants.

$$Y(t) = \sum_n H_n \exp i(2\pi f_n t - G_n)$$

Input:

LTERMS = Total integral number of terms to be produced. Can be calculated according to (3-2*ISUM)*(KOMPLX+1)*INT(E-BEGIN)/DT+1). where ISUM, KOMPLX, E, BEGIN, and DT are described below.

NGIVEN = Integral number of tidal constituents supplied by user. If ISUM = 0, NGIVEN must not be greater than 27, If ISUM = 1, NGIVEN must not be greater than 49. The constituents are designated according to Darwin, i.e., those separable in a year. The user may supply the constants from *any constituent* indicated in the table on page 62. The Darwin symbol, mean amplitude H and either the Greenwich epoch G, or the local epoch K are punched according to format A8, 2F8.3). The Darwin symbols must be exactly as in the table on page 62 and left justified in the A8 field. The epoch is given in degrees. The constituents can be placed in any order.

NUSED = Integral number of tidal constituents used in prediction series.
If ISUM = 0, NUSED must not be greater than 27.
= 1, NUSED must not be greater than 49.

INFER = If INFER = 1, the program attempts to infer the constants of all equilibrium (linear theory) dirunal and semidiurnal constituents which are not supplied by user. The diurnal inference is based on whichever of 01 or K1 is nearer the frequency. If only one of 01 and K1 has been supplied, all diurnal inferences are based on that one. The semidiurnal

Identification						DS*	Identification						DS*
0	0	1	0	0	0	SA	3	-4	3	0	0	0	2MK3
0	0	2	0	0	0	SSA	3	-3	3	0	0	0	M3
0	2	-2	0	0	0	MSF	3	-2	1	0	0	0	SØ3
0	2	0	0	0	0	MF	3	-2	3	0	0	0	MK3
1	-4	1	2	0	0	2Q1	3	0	0	0	0	0	S3
1	-4	3	0	0	0	SIGMA1	3	0	1	0	0	0	SK3
1	-3	1	1	0	0	Q1	4	-5	4	1	0	0	MN4
1	-3	3	-1	0	0	RØ1	4	-4	4	0	0	0	M4
1	-2	1	0	0	0	Ø1	4	-2	2	0	0	0	MS4
1	-1	1	1	0	0	M1	4	-2	4	0	0	0	MK4
1	0	-2	0	0	1	PI1	4	0	0	0	0	0	S4
1	0	-1	0	0	0	P1	6	-7	6	1	0	0	2MN6
1	0	0	0	0	0	S1	6	-6	6	0	0	0	M6
1	0	1	0	0	0	K1	6	-5	4	1	0	0	MSN6
1	1	1	-1	0	0	J1	6	-4	4	0	0	0	2MS6
1	2	1	0	0	0	ØØ1	6	-2	2	0	0	0	2SM6
2	-5	4	1	0	0	MNS2	6	0	0	0	0	0	S6
2	-4	2	2	0	0	2N2	8	-8	8	0	0	0	M8
2	-4	4	0	0	0	MU2	8	-7	6	1	0	0	2MSN8
2	-3	2	1	0	0	N2	8	-6	6	0	0	0	3MS8
2	-3	4	-1	0	0	NU2	8	-4	4	0	0	0	2(MS)8
2	-2	2	0	0	0	M2	8	0	0	0	0	0	S8
2	-1	2	-1	0	0	L2							
2	0	-1	0	0	1	T2							
2	0	0	0	0	0	S2							
2	0	2	0	0	0	K2							
2	2	-2	0	0	0	2SM2							

* Darwin Symbol
(0 means zero; Ø means the capital letter.)

constituents are based on whichever of N2, M2, or S2 is nearer the frequency which has been supplied. If it has not been supplied, the inference is based on M2 if it has been supplied. If constituent on which inferences are made is not supplied, no inference is made. If INFER = 0, non-supplied consitiuents are not to be inferred.

HMIN = Minimum amplitude of constituent to be taken into account in prediction. If H < HMIN, that constituent is omitted. Type real.

KAPPA = If KAPPA = 0, phases supplied are Greenwich Epoch.
If KAPPA = 1, phases are Local Epoch.

ELONG = The Greenwich East Longitude of the local epoch from which the phases are given. A correction is made by G = K - n*ELONG where n = 0, 1 or 2, designating species.

KOMPLX, If KOMPLX = 0, then a real series is produced depending on ISUM.
If KOMPLX = 1, then a series of alternation, real and imaginary terms are produced depending on ISUM. The imaginary term is in quadrature with the real term and has no geophysical significance. It is used for numeric stability in TIDWTS.

ISUM, If ISUM = 0, SERIES is composed of long-period, diurnal and semidiurnal subsums.
If ISUM = 1, SERIES is composed of the total sum of the constituents.

NOCARD, If NOCARD = 0, new consituents are being supplied by user for this call.
If NOCARD = 1, constituents supplied for the last previous call are to be used again.

ALLIST, If ALLIST = 0, output lists a table of constituents supplied by user, including Darwin symbol mean amplitude H, and the Greenwich or local epoc in degrees.
If ALLIST = 1, output includes, in addition to above table, the constituents used in the prediction, including Solar Doodson Number, Darwin symbol, frequency, Doodson coefficient, mean amplitude, Greenwich or local epoch, corrected amplitude, initial phase, and frequency in radians/interval and Nyquists.

All times are reckoned in hours from 0000 hours GMT 1 Jan 1900. See conversion table in front of tide section.

ET = End time of SERIES. Type real.

ST = Start time of SERIES. Type real, used in calculating nodal correction.

DT = Interval between successive terms in hours.

BEGIN = Actual start time of SERIES. Used when series is generated in pieces each LTERMS long starting at BEGIN.

A nodal correction is applied to the amplitude and phase, calculated at the mid-point of the series or at the middle of each year in the case of a series that is several years long.

Output:

SERIES - resultant series of length LTERMS. SERIES is real and consists of real term (KOMPLX = 0) or alternating real and imaginary parts when KOMPLX = 1. SERIES contains Y_0, Y_1, Y_2 corresponding to long period (note $Im(Y_0) = 0$), diurnal and semidiurnal subsums when ISUM = 0, and contain $Y_0 + Y_1 + Y_2 + Y_3 + Y_4 + Y_6 + Y_8$ when ISUM = 1. Where the subscript refers to species, i.e., long period, diurnal, semidiurnal, etc.

TIDWTS (REF,OBS,PH,SREF,EREF,SOBS,EOBS,DT,NCREF,PHW)

Function TIDWTS calculates the response weights which best fit the References REF to the observed OBS in the least squares sense. The matrix equation for weights W to predict V from the reference M,

$$[M_{qr}] \bullet [W_r^*] = [V_q]$$

is solved for $[W_r]$. The single suffix q or r denotes the double sequence (p,h) arranged in some order.

$$M_{qr} = \left\langle \{X_p(t+h)\}_q \ \{X_p(t+h)\}_r \right\rangle$$

$$V_q = \left\langle y(t)\{X_p(t+h)\}_q \right\rangle$$

The averages $\langle \ \rangle$ extend from ST to ET. Since Mqr is symetrical only one half of the diagonal matrix is actually computed. Inverting M_{qr} yields $[W_r]$.

REF consists of a merged complex series $X_p(t)$, p = 1, 2, 3,... usually generated by TIDPOT or TIDHAR. OBS is a single componenet real series y(t). The weights W_{ph} are found such that

$$\hat{y}(t) = \mathrm{Re}\{\sum_p \sum_h W_{ph}{}^* X_p(t+h)\}$$

is as close as possible to y(t) in the least squares sense. $X_p(t)$ and y(t) should have zero mean. The above solution is found for each reference series p.

Inputs:

REF = merged complex reference series as generated by TIDPOT or TIHAR or TIDRSP

OBS = real observation series for analysis

PH = array of p and h values for analysis. p referes to the complex component series of REF (e.g., p=2 refers to the second merged complex pair in REF). h refers to the lead (in hours) of reference series against the observed (e.g., h = -48 refers to a -48 hour lead of Ref over OBS). Note: h is usually in 48 hour multiples 0, $\pm$48 h $\pm$96 h, etc. p and h groups are separated by "4HSTOP". For termination of the sequence a double "4HSTOP" is used. See example page 69 for use.

SREF = start of reference series. It must be at most SOBS less the max h (SREF is less than SOBS by max lag).

EREF = end of reference series. It is larger than OBS and must be at least EOBS + largest h.

DT = time interval which must be same for REF and OBS in hours.

NCREF = number of component series in REF (e.g., if REF is a 3 component merged complex reference generated by TIDHAR then NCREF = 6).

Outputs:

PHW = is a quadruplet consisting of p, h, Re {w}, Im{w}.
The printout gives the predicted variance matrixes

$$g^{P}_{ph} = \mathrm{Re}\,\{W^{*}_{ph}\; V_{ph}\}\ ,\ \text{vectors}\ g^{P}_{p} = \sum_{h} g^{P}_{ph}$$

and groups totals $g^{P} = \sum_{p} g^{P}_{p}$, where g = 1, 2, 3,.... is the number of the p, h group. Next the recorded variance, $\sigma = \langle y^2 \rangle$, predicted variance $P = \langle \hat{y}^2 \rangle$ and residual variance $R = \sigma - P$ are listed. Finally p,h and the weights are listed.

TIDADM (PHW, LI, ISPEC, CMPNT)

Function TIDADM generates the complex linear admittance $Z_p(f)$ at frequencies depending on ISPEC, corresponding to the weights in PHW. PHW is of the form $[p,h,W_{ph}]$ as produced by TIDWTS. TIDADM has to be called separately for each desired p.

$$Z_p(f) = \sum_h W_{ph} e^{i\pi fh/12}$$

PHW = array of weights in form p, h, Re, Im, repeated as produced by TIDWTS.

LI = dimension of array PHW (a multiple of 4).

ISPEC = Species of tide analyzed for.

= 0, long period tides, f_s = 0.0366011 CPD
DELF = 0.0366011, f_E = 0.109803306 CPD
Lines analyzed - SSA, MSF, MF

= 1, diurnal tides, f_s = 0.892935 CPD
DELF = 0.0366011, f_E = 1.039339 CPD
Lines analyzed - Q1, O1, M1, P1, K1, J1.

= 2, semidirunal tides, f_s = 1.859071 CPD
DELF = 0.0366011, f_E = 2.005476.
Lines analyzed - 2N2, N2, M2, L2, S2, K2

where f_s = first frequency for admittance calculation
f_E = last frequency for admittance calculation
DELF = frequency interval for admittance calculation which is 1 cycle per month.

CMPNT = P componenet value from which the weights are selected.

At each frequency, $Z = X + iY = Re^{i\psi}$ is printed out in the following format:

CPY CPM CPD X Y R ψ

followd by the Darwin symbols and frequencies in cycles per solar day (CPD) of the most significant and most central tidal lines (if any) in the frequency band $f \pm \frac{1}{2}\delta f$.

TIDRSP (REF, LREF, PHW, LPHW, DT, SPRED, EPRED, SREF, EREF, JP, LJP, YPRED, LYPRED)

Function TIDRSP generates the tide by the response method assuming a reference series REF generated by TIDHAR.

$$Y_p(t) = \sum_h W_{ph} X_p(t+h)$$

where $X_p(t)$ is the reference series REF, and W_{ph} are the prediction weights PHW as generated by TIDWTS. The resultant series YPRED, Y_p , is a real series.

Inputs:

REF	=	6 component reference series generated by TIDHAR.
LREF	=	dimension of REF. Equals number of time points times 6.
PHW	=	array of response weights generated by TIDWTS consisting of P, H, real weight, imaginary weight.
LPHW	=	Dimension of PHW (a multiple of 4).
DT	=	interval between data points in hours.
SPRED	=	start time of resultant series (Geotime)
EPRED	=	end time of resultant series (Geotime)
SREF	=	start time of reference series (Geotime)
EREF	=	end time of reference series (Geotime)
JP	=	array of P values wanted in prediction (usually JP = 1, 2, and 3.)
LJP	=	dimension of array JP (usually 3)
LPRED	=	dimension of YPRED and equals (EPRED-SPRED)/DT+1

Outputs:

YPRED	=	array of predicted series.
TIDRSP	=	0, if the call was successful
	=	1, if an error occurred.

/

```
PROGRAM TIDE (INPUT, OUTPUT, TAPE1)
DIMENSION REF(10824), PH(12) OBS(1444), PHW(28), JP(3), YPRED(1444)

C       READ IN OBSERVED TIDAL PRESSURE
        READ (1) (OBS(I), I=1, 1444)
C       GENERATE REFERENCE SERIES AT CLAYOQUOT, B.C.
        SREF=611318.6
        EREF=611799.4
        DT=0.2666666
        BEGIN=SREF
        LTERMS=10824
        NGIVEN=27
        NUSED=27
        INFER=1
        HMIN=0.01
        KAPPA=1
        ELONG=232.9683
        KOMPLX=1
        ISUM=0
        NOCARD=0
        ALLIST=1
        T= TIDHAR(REF,LTERMS,NGIVEN,NUSED,INFER,HMIN,KAPPA,ELONG
       *KOMPLX,ISUM,NOCARD,ALLIST,EREF,SREF,DT,BEGIN)
        IF(T·NE·0.) GO TO 999
C       GENERATE RESPONSE WEIGHTS
        PH(1)=1.
        PH(2)=4HSTOP
        PH(3)=0.
        PH(4)=4HSTOP
        PH(5)=2.
        PH(6)=3.
        PH(7)=4HSTOP
        PH(8)=-48.
        PH(9)=0.
        PH(10)=48.
        PH(11)=4HSTOP
        PH(12)=4HSTOP
        SOBS=611366.6
        EOBS=611751.4
        NCREF=6
        W=TIDWTS(REF,OBS,PH,SREF,EREF,SOBS,EOBS,DT,NCREF,PHW
        IF (W·NE·).) TO TO 999
C       FIND THE ADMITTANCE
        LI=28
        ISPEC=0
        CMPNT=1.
        A=TIDADM(PHW,LI,ISPEC,CMPNT)
        ISPEC=1
        CMPNT=2.
```

```
      A=TIDADM(PHW,LI,ISPEC,CMPNT)
      ISPEC=2
      CMPNT=3.
      A=TIDADM(PHW,LI,ISPEC,CMPNT)
C           PREDICT SERIES FROM WEIGHTS AND REFERENCE
      JP(1)=1
      JP(2)=2
      JP(3)=3
      LJP=3
      LPRED=1444
      P=TIDRSP(REF,LTERMS,PHW,LI,DT,SOBS,EOBS,SREF,EREF,JP,LJP,YPRED,LPRED)
      IF(P·NE·0.) GO TO 999
      .
      .
      .
      .
      .
999   STOP
      END
```

Program TIDE analyses a series OBS for its tidal content using the response method. The steps are as follows:

1. Generate a reference series REF by the harmonic method from published tidal constants using the program TIDHAR. REF is a 6 component series.

2. The response weights are found to best fit REF to OBS in the least squares sense. The long period tide is analysed at 0 lag and the diurnal and semidiurnal at lags of -48, 0, and 48 hours.

3. The response weights PHW are then transformed to give the amplitude and phase at standard tidal frequencies by TIDADM.

4. A noise free predicted series is then generated by convolving the weights with the reference series to form YPRED. The function TIDRSP performs this operation.

Further analysis can be done to determine the signal-to-noise ratios, etc., by standard spectral techniques.

TIDFIL (FILTER, IT, IA)

Function TIDFIL generates one of three filters to discriminate against tidal frequencies.

Inputs:

IT = 1, produces weights for 25 hour averaging filter, i.e.,
FILTER(I) = 0.04

= 2, produces Doodson and Warburg 39 hour filter (see Doodson and Warburg, Admiralty Manual of tide, 1941, page 111.)

= 3, produces Groves 47 hour filter. See Groves, for D*47 filter weights, Numerical Filters for discrimination against tidal periodicities , Trans, AGU, 36:1073-1084, 1955.

IA = 0, produces one sided series of center term and right half for use with NCONVL.

1, produces series containing all weights.

Outputs:

FILTER = array of dimension depending on IT and IA containing the filter weights. (for length see below)

		IA 0	IA 1
IT	1	13	25
IT	2	20	39
IT	3	24	47

SUBROUTINE TIDSPC (SERIES, LOS, DT, CF, NAME)

Subroutine TIDSPC takes the fourier transform of SERIES at 15 diurnal and semidiurnal tidal frequencies. The transform is done at the exact tidal frequency which is not necessarily an even harmonic of the record length. numerically this produces fairly good amplitudes and fairly poor phases.

Inputs:

SERIES - time series of length LOS terms
LOS - length of array SERIES
DT - sample interval in hours
NAME - a name for printout purposes in A10 format
CF - normalization fraction by which spectral amplitudes are multiplied e.g., to correct for prefilter response.

Output:

A printed table. See example on following page.

NOTE: The mean is removed from SERIES thus altering the array SERIES. If a 10% cosine taper is applied to the series, then the amplitude response is 0.9, so the amplitude must be divided by 0.9 to produce the correct amplitude (CF = 1./0.9). See TAPER.

MEAN OF 1.143083E-01 REMOVED FROM SERIES COBB TIDE

SPECTRUM OF SERIES COBB TIDE AT TIDAL FREQUENCIES

TIDAL LINE	FREQUENCY CYCLES/DAY	AMPLITUDE UNITS	PHASE DEG	AMPLITUDE UNITS SQUARED	REAL(COS) UNITS	IMAG(SIN) UNITS
O1	.89324406	5.615636E+00	-114.3	3.153537E+01	-2.309581E+00	5.118711E+00
O1	.92953571	1.460954E+01	58.1	2.134386E+02	7.726631E+00	-1.239910E+01
M1	.96644626	3.636412E+00	-21.9	1.322349E+01	3.373214E+00	1.358277E+00
P1	.99726209	4.479441E+01	-51.0	2.006539E+03	2.817729E+01	3.482211E+01
K1	1.00273791	4.517917E+01	-17.3	2.041158E+03	4.313116E+01	1.344845E+01
J1	1.03902955	6.104675E+00	23.4	3.726705E+01	5.603474E+00	-2.422423E+00
2N2	1.85969032	7.288845E+00	-153.6	5.312726E+01	-6.528660E+00	3.240967E+00
MU2	1.86454723	6.180595E+00	-112.8	3.819975E+01	-2.392013E+00	5.698950E+00
N2	1.89598197	1.014021E+01	150.3	1.028240E+02	-8.804230E+00	-5.030854E+00
NU2	1.90083887	1.456930E+01	141.8	2.122646E+02	-1.144755E+01	-9.012112E+00
M2	1.93227361	7.984334E+01	-72.3	6.374958E+03	2.421921E+01	7.608146E+01
L2	1.96856526	1.121722E+01	-39.1	1.258260E+02	8.708443E+00	7.070294E+00
T2	1.99726222	1.636781E+01	82.4	2.679051E+02	2.156292E+00	-1.622515E+01
S2	2.00000000	1.517582E+01	90.3	2.303057E+02	-8.954918E-02	-1.517556E+01
K2	2.00547582	1.330081E+01	108.2	1.769116E+02	-4.151381E+00	-1.263636E+01

FOR UNCLASSIFIED TECHNICAL REPORTS, REPRINTS, & FINAL REPORTS
PUBLISHED BY OCEANOGRAPHIC CONTRACTORS
OF THE OCEAN SCIENCE & TECHNOLOGY DIVISION
OF THE OFFICE OF NAVAL RESEARCH
(REVISED SEPTEMBER, 1973)

1 Director of Defense Research and Engineering
Office of the Secretary of Defense
Washington, D.C. 20301
ATTN: Office Assistant Director (Research)

12 Defense Documentation Center
Cameron Station
Alexandria, Virginia 22314

Office of Naval Research
Department of the Navy
Arlington, Virginia 22217
3 ATTN: Code 480
1 ATTN: Code 460
1 ATTN: Code 102-OS

1 Cognizant ONR Branch Office

1 ONR Resident Representative

Director
Naval Research Laboratory
Washington, D.C. 20390
6 ATTN: Library, Code 2029 (ONRL)
6 ATTN: Library, Code 2620

Commander
Naval Oceanographic Office
Washington, D.C. 20390
1 ATTN: Code 1640
1 ATTN: Code 70

1 Director
National Oceanographic Data Center
National Oceanic & Atmospheric Administration
Rockville, Maryland 20882